Detection of the Sunyaev-Zeldovich effect through Deep Learning techniques

Detección del efecto Sunyaev-Zeldovich mediante técnicas de Deep Learning

Trabajo de Fin de Máster
para acceder al

Máster en Física de Partículas y del Cosmos

Autor: Jose María Palencia Sainz

Directores: Patricio Vielva Martínez

Patricia Diego Palazuelos

July - 2021

Index

Acknowledgements

Este trabajo supone el final de mis estudios de física. Querría agradecer a todas esas personas, profesores y alumnos, que me han acompañado durante mis cuatro años de carrera y este último curso en el máster.

Mención especial a mis directores del proyecto, Patricio y Patricia. Gracias por el apoyo, ayudandome en los momentos de mayor estrés en el trabajo, y gracias por la experiencia y sabiduría compartidas conmigo en este apasionante campo. Me quedo con mi introducción a las redes neuronales, así como mi mejor comprensión del FCM y los fenómenos involucrados en su medición que me habéis aportado. Mención importante a Diego, mi director de TFG, quién me introduzco en la maravillosa ciencia del FCM y la separación de componentes. Entre todos me habeís ayudado a dar un paso más en mi camino a la investigación profesional, entendiendo mejor el día a día de un investigador en física y los retos a los que este se enfrenta.

Este trabajo ha sido parte de una beca de colaboración de iniciación a la investigación, JAE Intro-SOMdM 2020, fundada por el CSIC. Gracias por la oportunidad, y gracias al Grupo de Cosmología Observacional e Instrumental del IFCA por ofrecerme el trabajo.

Finalmente, gracías a mis amigos y familia, en especial mis padres y mi hermana, por estar ahí en el día a día e interesarse en mi trabajo, así como por su preocupación por mí, especialmente en esos días finales de mayor estrés antes de la entrega. Me embarco ahora en una nueva etapa, una beca de doctorado concedida por el Gobierno de España, durante los próximos cuatro años me embarcaré en una aventura en la detección de agujeros negros primordiales en el IFCA, procuraré utilizar todo lo aprendido estos cinco años de la mejor forma.

Abstract / Resumen

The component separation of the microwave sky (i.e. recovering the different galactic foregrounds and the Cosmic Microwave Background, CMB) has important implications in both cosmology and astrophysics. It allows an accurate characterization of the CMB and the foregrounds, hence, a proper analysis of the cosmological parameters and tests of several astrophysical theories. In this work, we present a new approach to the detection of the Sunyaev-Zeldovich effect (a secondary anisotropy of the CMB photons caused by the intra-galaxy clusters electron gas) using convolutional neural networks, CNNs, on multi-frequency maps of the experiments dedicated to the CMB detection. We want to set the basis for a more detailed work, comparing the efficacy and efficiency of this new detection method with the usual multi-frequency filters methods.

In this project we have trained a CNN with a data set, from simulations of the CMB, SZ emission, and Gaussian noise. This first model has successfully identified the frequency dependence of the SZ, allowing its detection. However, we have noticed an strange behaviour between the output of the network and the flux of the sources. We have some theories about it.

Keywords: Component separation, Sunyaev-Zeldovich, Galaxy clusters, Cosmic Microwave Background, Deep Learning, Convolutional Neural Networks, Cosmology.

La separación de componentes en el cielo de microondas (es decir, recuperar los distintos fondos galácticos y la radiación de Fondo Cósmico de Microondas, FCM) tiene importantes implicaciones en la cosmología y astrofísica. Permite una precisa caracterización del FCM y de los distintos fondos, y con ello, un correcto análisis de los parámetros cosmológicos y la prueba de distintas teorías astrofísicas. En este trabajo, presentamos una nueva alternativa a la detección del efecto Sunyaev-Zeldovich (una anisotropía secundaria de los fotones del FCM causada por los electrones del gas intra-cumular en los grandes cúmulos de galaxias) usando redes neuronales convolucionadas, CNNs (por sus siglas en inglés), en mapas multi-frecuencias de experimentos dedicados a la detección del FCM.. Buscamos sentar las bases para un trabajo más detallado, comparando la eficiencia y eficacia de este nuevo método de detección con los usuales metodos de filtros multi-frecuencias.

En estre poyecto hemos entrenado una CNN, con simulaciones del FCM, la emisión SZ, y ruido Gaussiano. Este primer modelo ha logrado identificar la dependencia frecuencial del effecto SZ, permitiendo su detección. Sin embargo, hemos notado un extraño comportamiento entre la salida de la red y el flujo de las fuentes, aunque tenemos algunas teorías sobre ello.

Palabras clave: Separación de componentes, Sunyaev-Zeldovich, cúmulos de galaxias, Fondo Cósmico de Microondas, Deep Learning, redes neuronales convolucionadas, cosmología.

Chapter 1

Introduction

In this chapter a general discussion about the importance of the CMB science will be given. We will also include an general overview of the main contaminant foregrounds of the CMB, and the component separation in the context of the CMB. We will end with a relatively detailed introduction of the Sunyaev-Zeldovich effect.

1.1 Cosmic Microwave Background

Roughly 14 billion years ago, the Universe was a hot and dense aggregation of matter and radiation. Initially, barionic and cold dark matter were in the form of a hot plasma coupled to radiation. This aggregate of matter and radiation was expanding, lowering the temperature of the clump. Around 375 000 years after the end of inflation, the temperature was cold enough to allow the electron-proton recombination into the first atoms. At this point, radiation was no longer coupled to matter, and the Universe was no longer opaque, but transparent, setting radiation free. The emitted photons have freely travelled from and through all the Universe until eventually reaching us today. This radiation emitted at the time know as recombination, is well known in modern cosmology and astrophysics as the Cosmic Microwave Background radiation, or CMB, indeed it is a pillar to modern cosmology.

The CMB radiation perfectly fits a blackbody spectrum. In fact, it it the best fit we have encountered in nature. It was originally hotter but, as a consequence of the expansion of the Universe, its blackbody temperature dropped. Now it lies at $T_0 = 2.725 \pm 0.001$ K [1], which makes this blackbody radiation peaks at the microwaves range.

The CMB is nowadays one of the main supporters of the standard cosmological model (ΛCDM). This model is an Einstein's general theory of relativity description of the Universe, it includes the theory of the Big-Bang and the cosmological principle, i.e., the Universe is homogeneous and isotropic (only at the large scales), meaning that there is no privileged observer. The model also assumes an early, period described by cosmic inflation, seeding the primordial energy density perturbations.

The CMB was casually discovered by Penzias and Wilson in 1964 [2], since that moment cosmologists

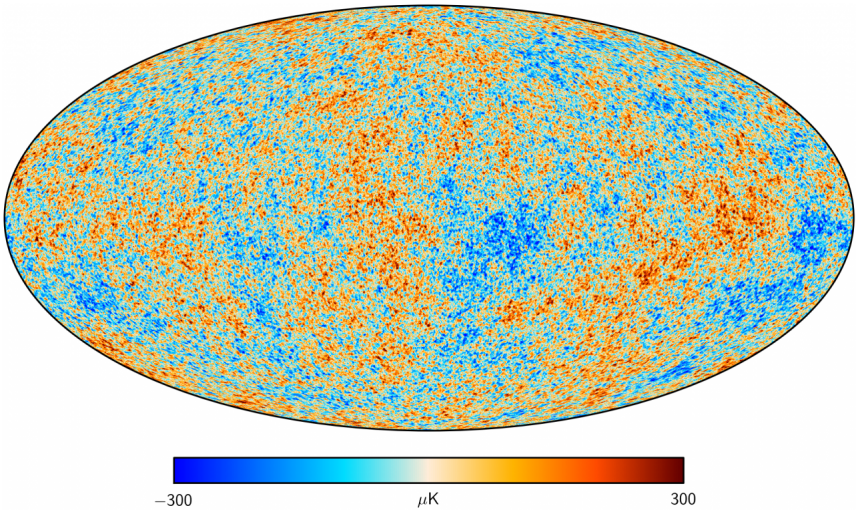

Figure 1.1: All-sky map of the CMB temperature anisotropies.[1]

have put an enormous amount of work in the characterization of this radiation. Despite been homogeneous, the CMB has tiny temperature anisotropies of the order of $10^{-6} - 10^{-5}$K. It also has a dipole caused by the relative motion of the Sun (doppler effect with respect to the CMB rest of frame), this has been well characterized and can be subtracted. The remaining anisotropies come from the initial inhomogeneities of the Universe at the time of recombination.. It is believed that these inhomogeneities evolved from the inflaton fluctuations. Some astrophysical and cosmological effects impact this initial anisotropies (see [1] and [3] for more information on CMB anisotropies). A precise analysis of the CMB anisotropies provides estimations of many cosmological parameters such as, matter and radiation energy densities, the age of the Universe, the Hubble constant, or the cosmological constant (related to Dark Energy) [1]. Since 1965 several ground based, balloon borne and spatial satellites have studied the microwave sky, both grounded and spatial antennas. Since the atmosphere absorption window covers most of the microwave range, mainly due to the H_20 molecules in the air, the best results come from the spatial missions, such as COBE[1](1989), WMAP[2](2001) and Planck[3](2009, [1]), with increasing precision. Future missions and analysis will characterized the CMB with higher precision which will give us, combined with other equally important measurements, a depper understanding of the Cosmos.

Anisotropies and angular power spectrum

The CMB temperature anisotropies are studied as temperature fluctuation of a scalar field [4]. The normalized temperature in the direction $\hat{\mathbf{n}}$ on the celestial sphere, Θ, is defined as $\Theta(\hat{\mathbf{n}})= \frac{\Delta T(\hat{\mathbf{n}})}{\langle T \rangle}$, where $\Delta T(\hat{\mathbf{n}}) = T(\hat{\mathbf{n}}) - T_0$. This temperature field can be expressed as a multipole decomposition in terms of the spherical harmonics $Y_{\ell m}$:

$$a_{\ell m} = \int \Theta(\hat{\mathbf{n}}) Y^*_{\ell m}(\hat{\mathbf{n}}) d\Omega, \tag{1.1}$$

[1]https://science.nasa.gov/missions/cobe
[2]https://map.gsfc.nasa.gov/
[3]https://www.cosmos.esa.int/web/planck

where the integral covers the whole sphere. Inversely:

$$\Theta(\hat{\mathbf{n}}) = \sum_{\ell=0}^{\ell_{max}} \sum_{m=-\ell}^{+\ell} a_{\ell m} Y_{\ell m}. \tag{1.2}$$

If this temperature field would arises from a Gaussian random field, as it is though, the field will be, then, fully characterized by its angular power spectra, C_ℓ, $\left\langle a^*_{\ell m} a_{\ell' m'} \right\rangle = \delta_{\ell\ell'}\delta_{mm'}C_\ell$ (the brackets denotes an average over a sky with the same cosmology). The order, m, describes the angular orientation of a mode, whereas the multipole, ℓ, on the other side, describes the angular size of the mode. As the Universe is expected to be isotropic, not preferred direction, the power spectrum is then independent of m. As it should be independent of m, the best estimate of C_ℓ is the average over m.

At each multipole ℓ, there are only (2ℓ+1) modes to detect the power, this fundamental limit is known as the cosmic variance:

$$\frac{\Delta C_\ell}{C_\ell} = \sqrt{\frac{2}{2\ell+1}}. \tag{1.3}$$

The full uncertainty in the power at each multipole arise from the instrumental noise, a finite beam resolution, and the observation of a finite fraction of the full sky [5]:

$$\Delta C_\ell = \sqrt{\frac{2}{2l+1}} \frac{1}{\sqrt{f_{sky}}} \left(C_\ell + N_\ell b_\ell^{-2}\right), \tag{1.4}$$

where, f_{sky} represents the observed fraction of the sky, N_ℓ is the instrumental noise angular power spectra, given by $N_\ell = \frac{\sigma^2}{4\pi}$ (with σ the noise sensitivity per steradian), and b_ℓ is the instrumental beam window function. It is usually described by a Gaussian function that, in harmonic space, reads as $b_l = e^{-\ell(\ell+1)\sigma_b^2/2}$, where σ_b (in radians) is the Gaussian bean dispersion. For example the WMAP satellite had a beam of $\sim$ 0.2°, this limited the sensibility to below $l \approx 600$, for Planck, which had a much smaller resolution, the power sensitivity was up to $\ell \approx$2500, giving a greater angular resolution. The C_ℓ coefficients contain the required information needed for the characterization of the, hence, allowing the determination of several cosmological parameters, see Figure 1.2.

The CMB radiation is also polarized. In fact, CMB polarization is one of the most active fields of study in cosmology, notably in relation to the so-called B-mode, which is seen as a capital observable to provide a fundamental confirmation of the inflation model. However, we will work only in temperature as a first step on the Sunyaev-Zeldovich detection with CNNs.

1.2 Foregrounds

Observing the microwave sky means that not only the CMB is observed, but also different sources of radiation that emits at the microwave frequencies. Complex pipelines are required in order to obtain a clean image of the CMB. To do so, it is mandatory to understand deeply these foregrounds that contaminate the CMB (also this can be looked at in the other direction. We might want a foreground and the CMB contaminates the image [6]), with this information a subtraction can be applied obtaining the CMB on one hand, and the foregrounds on the other. This is known as the component separation. There are two main

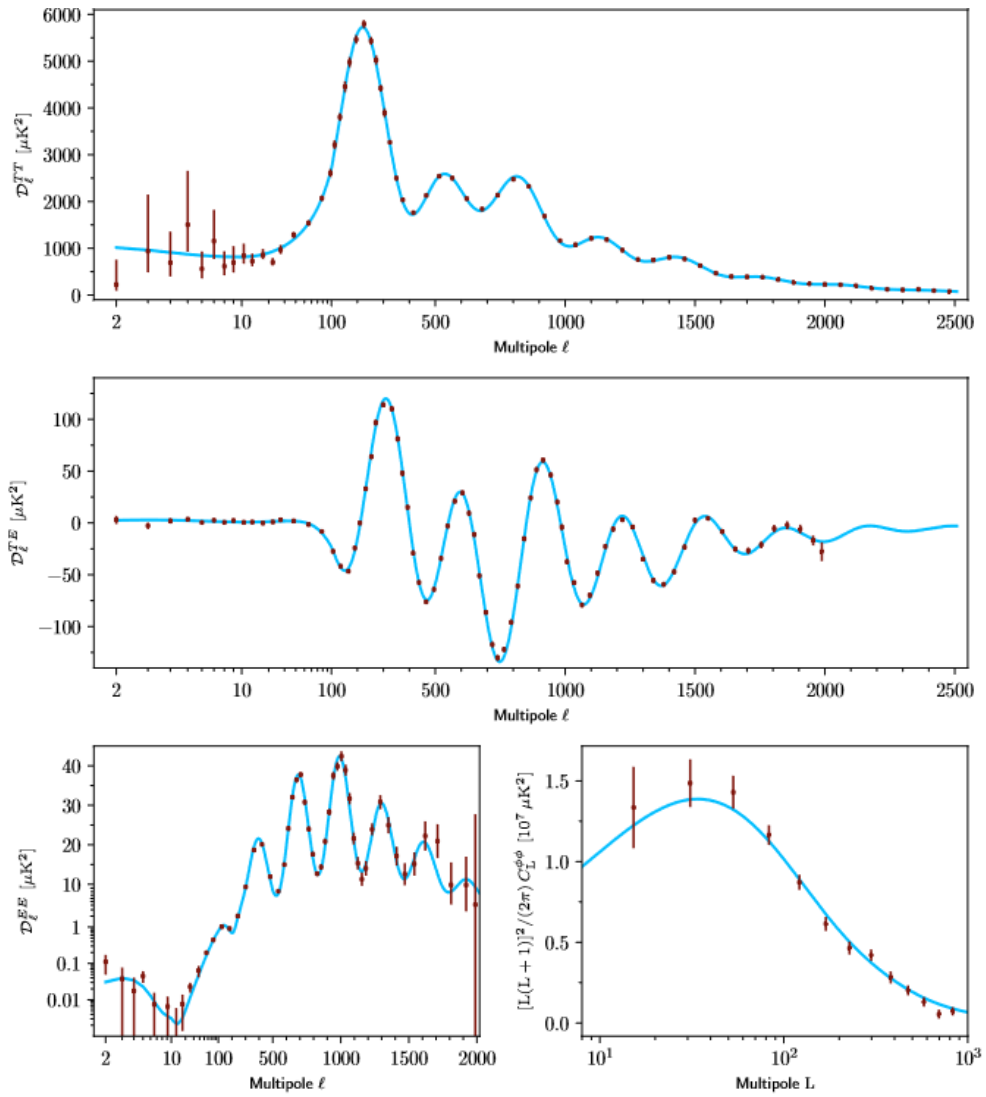

Figure 1.2: Planck CMB angular power spectra from [1]. Foreground-substracted, frequency-averaged temperature angular power spectra (top), temperature-polarization cross-angular power spectrum (middle), E polarization angular power spectrum. (bottom left), and lensing potential (bottom right). Most of the ΛCDM cosmological model information that can be derived from Planck's data is contained here (see [1] for more details).

types of foregrounds: galactic and extragalactic. Extragalactic foregrounds appear highly homogeneous through the sky, while galactic contaminants, contribute mainly in the galactic plane. Foregrounds are usually classified as compact of diffuse, the diffuse foregrounds have not a clear border while compact foregrounds have. This distinction depends somehow on the angular resolutions of our telescopes.

Many of these effects appear in our galaxy, mainly contaminating the galactic plane, and also at higher latitudes, this effect peaks at different frequencies and depends on the foreground. Many galactic foregrounds are observer as diffused sources due to their proximity to us, but there are also compact galaxy foregrounds such as the ultra-compact H_{II} regions. On the other hand extragalactic foregrounds also present the same emissions acting in our galaxy, slightly modified due to the specific galaxy characteristics, but seen as compact objects as they are farther away. There are also near extragalactic sources that can be

resolved and seen as diffuse. These foregrounds are practically homogeneously distributed.

The foregrounds, in our galaxy, are more intense in the galactic plane and their intensity decreases with the galactic latitude, b, according to a cosecant law [7] (the optical depth, τ, scaling is proportional to $1/\sin|b|$). This emission carries important information about the interstellar medium, ISM.

The more reasonable solution to avoid as much contaminants as possible in our CMB observations would be measuring the CMB at higher latitudes, near the galactic poles, far from the Milky Way, however, we would be compromising the large scale information. At lower latitudes we have many compact sources seen as bright spots. The only solution is to detect the contaminants and masked them, which is a hard to perform. A representation of how important are the foreground intensities with respect to the CMB intensity at different frequencies is shown in figure 1.3. An overview of these foregrounds is given in [8] and [7].

Galactic synchrotron emission is the most important diffuse galactic foreground at radio frequencies. It is produced by relativistic cosmic rays. This cosmic rays are accelerated by magnetic fields. It is the dominant contaminant of the CMB for frequencies below 80GHz [9]. The intensity of this emission depends on the cosmic ray density, and the strength of the magnetic field perpendicular to the line of sight. In the simplest model, for the radio-microwave regime, the density follows a power law, and so does the frequency dependence.

Free-Free emission is an other galactic radiation which is produced by electrons interacting with ions in the interstellar medium. It is fainter than other emissions such as synchrotron or thermal dust, except in regions of active star-formation. It depends slightly on the electron temperature, and varies slowly with the frequency [10]. Free-Free radiation is particularly relevant at a small range of frequencies around 30-50 GHz.

An excess amount of emission was found in the 10-100 GHz range and it was called, the anomalous microwave emission [11]. Above 20 GHz its spectrum decreased with frequency similar to synchrotron, but the spatial distribution was similar to the distribution of thermal dust. The proposed model was small spinning dust particles producing the radiation. Below 20 GHz, the spectrum flattens and, eventually decreases.

Heated dust grains emit radiation, it is known as the galactic thermal dust emission, and is the one that dominates at frequencies higher than 80GHz. The analysis of this radiation is extremely complex, it requires to know the composition of the dust, its sizes, the spatial distribution and many other parameters. It can be simplified assuming that, there are two population of grains well mixed and heated by the same radiation; both populations have constant optical properties; and the mass ratio remains the same. With this the emission can be modelled as the sum of two grey body radiations [12].

Molecular emission lines are seen in dense molecular clouds and they constitute the last galactic foreground in relevance. The excitation/de-excitation of certain molecules have an spectral shaped well

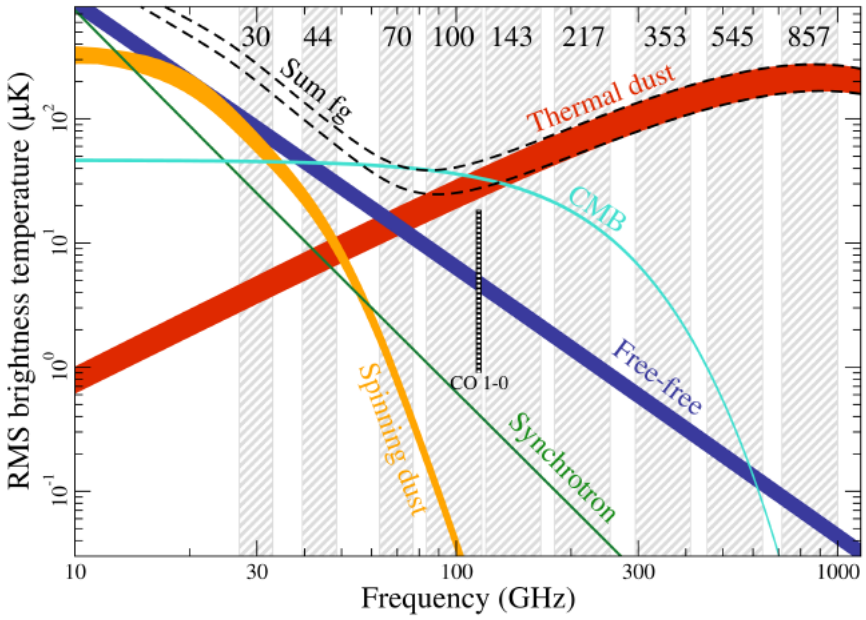

Figure 1.3: Intensity of the main foregrounds in the submillimiter sky given in brightness temperature (temperature of a fictional blackbody to duplicate the emitted radiation) at different frequencies, compared to the CMB radiation intensity. The grey vertical bars represent the coverage of Planck at each channel. Source [1].

defined. The most important lines at the CMB frequencies are the ^{12}CO lines (115.27 GHz, 230.54 GHz, and 345.80 GHz), although, there are other emission lines, less important, such as ^{13}CO (110.20 GHz), HCN (88.63 GHz), or HCO^+ (89.19 GHz).

Extragalactic radio sources appears as point like contaminants. They are mainly active galactic nuclei (AGNs), quasars or radio galaxies, depending on the orientation towards the observer, they can also be pulsars or certain nebulas. Radios sources spectrum emission is often approximated to a power law. They also appear as point like objects.

Extragalactic far infrared sources are identified as ultra-compact H_{II} regions (ionized hydrogen in star forming regions), far infrared galaxies, or heated molecular clouds, to mention some. All but the H_{II} regions can be characterized as modified blackbody emissions, while the H_{II} is an emission line. Also the ultra-compact H_{II} are also candidates to galactic point sources.

1.3 Sunyaev-Zeldovich effect

A hot gas of electrons can be found within the largest structures in our Universe, the galaxy clusters. This hot electron gas interacts with the incoming photons of the CMB; by an inverse compton scattering the CMB photons energy is boosted to higher energies, this creates secondary anisotropies in the CMB radiation. This effect is known as the thermal Sunyaev-Zeldovich (tSZ) effect as it was first described by Sunyaev and Zeldovich in 1972 [13] (see [14] a more modern description). There is also another Sunyaev-Zeldovich effect, the kinetic Sunyaev-Zeldovich effect (kSZ), produced by photons interacting

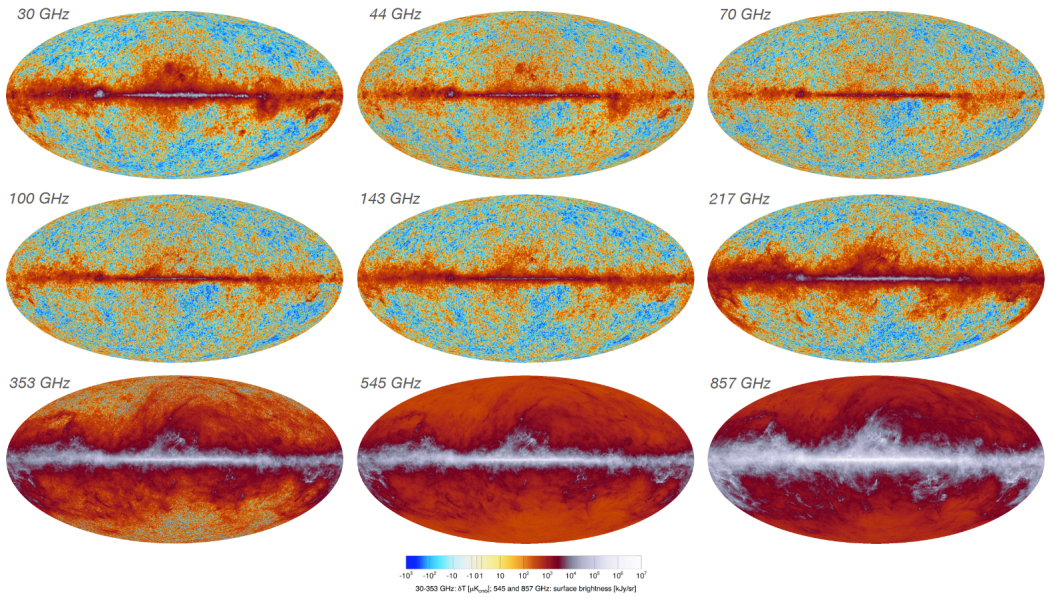

Figure 1.4: Emission fluctuations for the submillimeter sky, after solar dipole removal, at each of the nine Planck's channels. The fluctuations are expressed as equivalent temperature variations for the seven lowest frequencies, the highest two frequencies are expressed in more conventional units. Source [1].

with an electron gas with a peculiar velocity along the line of sight.

1. **Thermal SZ:**

   The electron population follows a Fermi distribution with a temperature $T_e \gg T_{CMB}$ (typical order corresponds to a few keV). The probability of a CMB photon interacting with a high energy electron, from the intracluster medium (ICM), is $\approx 1\%$, and the preferential boost is roughly $k_b T_e / m_e c^2$, where k_b is the Boltzmann constant, m_e the electron mass, and c is the speed of light. This distortion corresponds to ≤ 1 mK.

   This effect creates a decrease in the intensity of the background below 218 GHz, and an increase in higher frequencies.

   This distortion can be expressed as a change in temperature ΔT_{SZ}:

   $$\frac{\Delta T_{SZ}}{T_{CMB}} = f(x) y_c = \left(x \frac{e^x + 1}{e^x - 1} + 4 \right) (1 + \delta_{SZ}(x, T_e)) y_c, \tag{1.5}$$

   where $f(x)$ is the frequency dependence at dimensionless frequency $x \equiv \frac{h\nu}{k_b T_{CMB}}$, h is the Planck constant, k_b is the Boltzmann constant, and δ_{SZ} is a relativistic correction. In the non-relativistic limit f(x) $\rightarrow$ 2. y_c is the Compton parameter:

   $$y_c = \int n_e \frac{k_b T_e}{m_e c^2} \sigma_T dl, \tag{1.6}$$

   where n_e is the electron number density, and σ_T is the Thompson cross section. For isothermal clusters y_c equals the optical depth, τ_e, times the fractional energy gain per scattering. This temperature shift is independent of the redshift z, this means that the SZ is a powerful method to explore the Universe at higher redshifts.

The SZ effect can be also given in terms of intensity as:

$$\Delta I_{SZ} = g(x) I_0 y = \frac{x^4 e^x}{(e^x - 1)^2} f(x) I_0 y. \tag{1.7}$$

where $I_0 = 2(k_b T_{CMB})^3/(hc)^2$.

Typical massive galaxy clusters have a gas temperatures around $k_B T_e \sim 10$ keV. It is expected that temperature scale with mass as $T_e \propto M^{2/3}$. For this massive clusters, electrons became relativistic and small corrections must be applied [15]. For a massive cluster of $k_B T_e \sim 10$ keV, the relativistic correction is of the order of a few percent in the Rayleigh-Jeans (RJ) portion of the spectrum (low frequencies).

The SZ is the integrated effect of the electrons in the cluster over the solid angle, all the electrons weighted by their temperature. Integrating over the solid angle $d\Omega = dA/D_A^2$:

$$\int \Delta T_{SZ} d\Omega \propto \frac{M \langle T_e \rangle}{D_A^2}, \tag{1.8}$$

where M is the total mass of the gas, and $D_A(z)$ is the angular distance, almost flat at higher redshifts. The most important features of the tSZ are: 1) creates small distortion of the CMB spectra (~ 1 mK, 2) it has a sign change at 218 GHz, 3) it is independent of the redshift, and 4) the integrated SZ effect is proportional to the mass weighted by the electron temperature.

2. **Kinetic SZ:**

   When a cluster is moving in the radial direction, there will be a Doppler effect. This causes an additional spectra distortion on the already scattered photons of the CMB, in the relativistic limit. This effects happens if the cluster is moving with respect to the CMB rest frame, but it is only noticeable by us if there is a motion in the radial direction, v_r. This second distortion is the kSZ effect. In the non-relativistic limit:

$$\frac{\Delta T_{SZ}}{T_{CMB}} = -\tau_e \left(\frac{v_r}{c} \right), \tag{1.9}$$

   the CMB spectrum remains a blackbody but with a slightly different temperature, lower/higher for positive/negative radial velocities [16].

   Relativistic perturbation in the kSZ came from Lorentz boots to the electrons by the net velocity of the ensemble [17]. The most important term is of the order of $(k_B T_e/m_e c^2)(v_r/c)$. For a typical massive cluster, $k_B T_e = 10$ keV and $v_r = 1000$ kms^{-1}, this correction is an 8% of the non-relativistic term.

The SZ effect is nowadays one of the most powerful tools we have for the detection of galaxy clusters. As it is redshift independent, it is extremely useful for the analysis of the high redshift Universe. Although we use the SZ for galaxy clusters detection, the whole analysis of these clusters requires a multi-frequency survey, especially in the x-ray and optical range.

In summary, the SZ can be seen as a decrements of the CMB flux in the lower frequencies, and as an increment in the higher frequencies, it behaves as secondary anisotropies, a contaminant that we want to

take out from our CMB analysis. In order to do so, we need to identify the position of this anisotropies, to take them out of the images, also the SZ catalogue is important for galaxy clusters studies. Since the count distribution provides information about dark matter and the matter growths and density fluctuations. In this frame we will develop our work, a tool for the SZ detection on CMB images using artificial intelligence, we propose the use of CNNs as an alternative to the usual methods based on filters.

Finally the SZ effect and other foreground also change the polarization of the CMB, however, in this initial work we will be working only with temperatures, to know more about how this foregrounds and specially the SZ effect affect the CMB polarization see [7].

Chapter 2

Methodology

This chapter will present the algorithms developed for the detection of the SZ effect in the CMB. First we will describe the simulation of the sky (CMB + noise + SZ + foregrounds), and its patching (projection on the plane) in order to get the images that will be given to the network. The second part will provide the foundations of CNNs and the architecture of our network. Third section will detail the detection process, how do we use the network's output, the detection criteria, and how do we evaluate the goodness of the method. All computations have been perform on the Python programming language [18]. Our methodology can be summarize as follows:

1. Generate precise CMB simulations according to ΛCDM cosmology model.
2. Generate accurate SZ foreground maps.
3. Add both foregrounds and simulate Planck's experimental response (noise and resolution per channel).
4. Divide the sphere in patches without redundant information (no patches overlapping).
5. Create a model using a CNN with validation and training sets.
6. Use the trained model on different data and study the results.

2.1 Simulations

The reason behind the use of simulations instead of real data is that we can customize the data (i.e., selecting the foregrounds that we want, know the position of each cluster, create a CMB according to a given cosmological model etc...), and, therefore, to have a better performance of the tool. In addition we can generate more data than the real, we can generate different realizations of the CMB completely random, for instance we have simulated three different skies, this helps studying the statistical significance of the data.

We will work with simulation in spherical maps. HEALPIX[1], a software designed for the WMAP [19], provides a pixelisation of the celestial sphere. It is the standard software in cosmology to make calculations on the sphere. HEALPIX is implemented in C, C++, Fortran90, IDL, Java and Python, the last implementation is known as healpy [20].

HEALPIX tessellates the sphere into curvilinear quadrilaterals. The base resolution consists of 12 pixels, and it grows by the division of each pixel into four new pixels. Figure 2.1 illustrates, in the clockwise direction starting from the upper left, how the sphere is divided into 12, 48, 192, and 768 pixels. HEALPIX pixels, at the same resolution, cover the exact same area of the sphere. The last essential property of the HEALPIX pixels is that they are distributed into isolatitude lines, this is essential for harmonic analysis applications (e.g. calculating the spherical harmonic coefficients), it makes the computational cost scale as $N^{1/2}$ instead of scaling as N. Every pixel is identified by an integer, N_{pix}, in the range [0, $12N_{side}^2$-1]. The resolution that we have used is N_{side}=2048 which corresponds to 50 331 648 pixels, the resolution of the map (the area of each pixel) can be easily calculated, for this particular N_{side} is 1.718×1.718 arcmin^2. As explained in chapter 1, a microwave sky observation at a given frequency

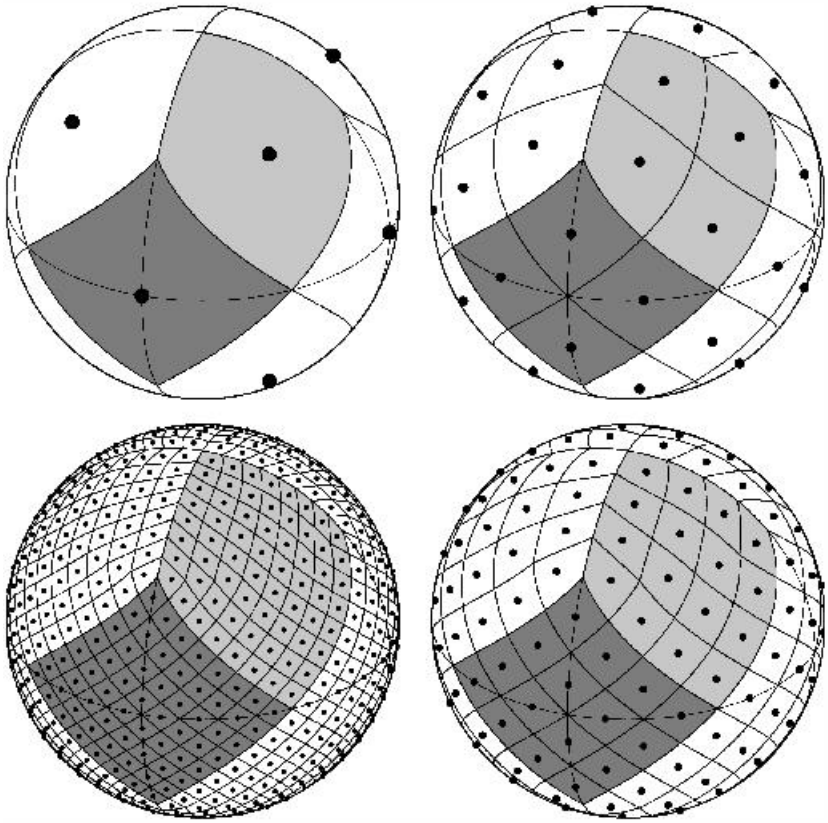

Figure 2.1: Representation of the HELAPIX tessellation of the sphere using 12, 48, 192 and 768 pixels, from upper-left to bottom left in a clockwise direction. Source: https://healpix.sourceforge.io/

includes several components. In our simulations, these components are produced independently to be later combined in to the total map. In our simple model our components are the CMB, instrumental noise,

[1]https://healpix.sourceforge.io/

and the SZ signal. A later test was realized on sky including the galactic and extragalactic foregrounds. To see further information on HEALPIX see [21].

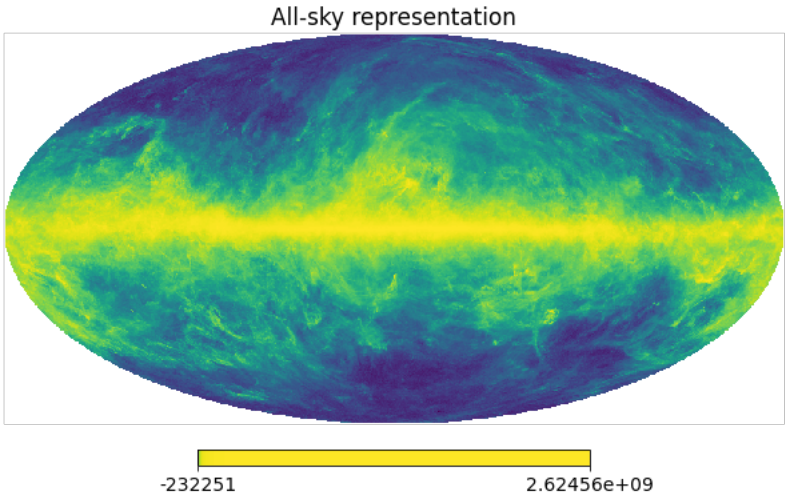

Figure 2.2: Projection, provided by HEALPIX, of the combined simulation of the CMB + Gaussian noise + galactic foregrounds This is given in units of temperature fluctuation, with respect to the CMB temperature, in μK.

To simulate the instrumental effects we have used the data from the table 2.1. We have added to each pixel, at each frequency map, a random number from a Gaussian distribution with zero mean and standard deviation equal to the temperature noise of the instrument (re-scaled from 1°to our pixel area). Once we have the whole sky we want to study, we convolve each map by a gaussian beam of FWHM as the channel resolution, for this we have used the healpy function `smoothing`.

Table 2.1: Main characteristics of Planck channels.

	Frequency [GHz]								
Property	30	44	70	100	143	217	353	545	857
Frequency [GHz][a]	28.4	44.1	70.4	100	143	217	353	545	857
Effective beam FWHM [arcmin][b] ..	32.29	27.94	13.08	9.66	7.22	4.90	4.92	4.67	4.22
Temperature noise level [μK_{CMB} deg][c]	2.5	2.7	3.5	1.29	0.55	0.78	2.56		
[kJy sr^{-1} deg][c]								0.78	0.72

[a] Center frequency for the LFI channels. Identifier for HFI channels.
[b] Mean FWHM of the elliptical gaussian fit to the effective beam.
[c] Noise intensity scaled to 1°. White noise assumed.

2.1.1 CMB simulation

To generate the CMB we have used the Code for Anisotropies in the Microwave Background or CAMB[2] (see [22] and [23]). This software implemented in Python and Fortran is an optimized code that among its features includes the calculation of the CMB, lensing, source count, and dark-age 21cm angular power spectra. It allows different cosmological models and parameters. It also supports closed, open and flat universe models. It has many other features but we are interested in the CMB angular power spectra calculation.

[2]http://camb.info

We have used the Planck best fit cosmological parameters [1] to calculate with CAMB the CMB ΛCDM power spectrum (C_ℓ). With this power spectrum we use the healpy function **synfast**, it creates a map of a given N_{side} with random fluctuations. It computes the spherical harmonic coefficients $a_{\ell m}$, with $2 \leq \ell \leq \ell_{max}$, and $-\ell \leq m \leq \ell$. There is no correlation between these coefficients and so they can be generated independently for a fixed ℓ. They are generated by a Gaussian distribution with zero mean, (remember that C_ℓ is the variance). From the same spectra this function creates a new random simulation of the map, they are different skies with the same angular power spectra.

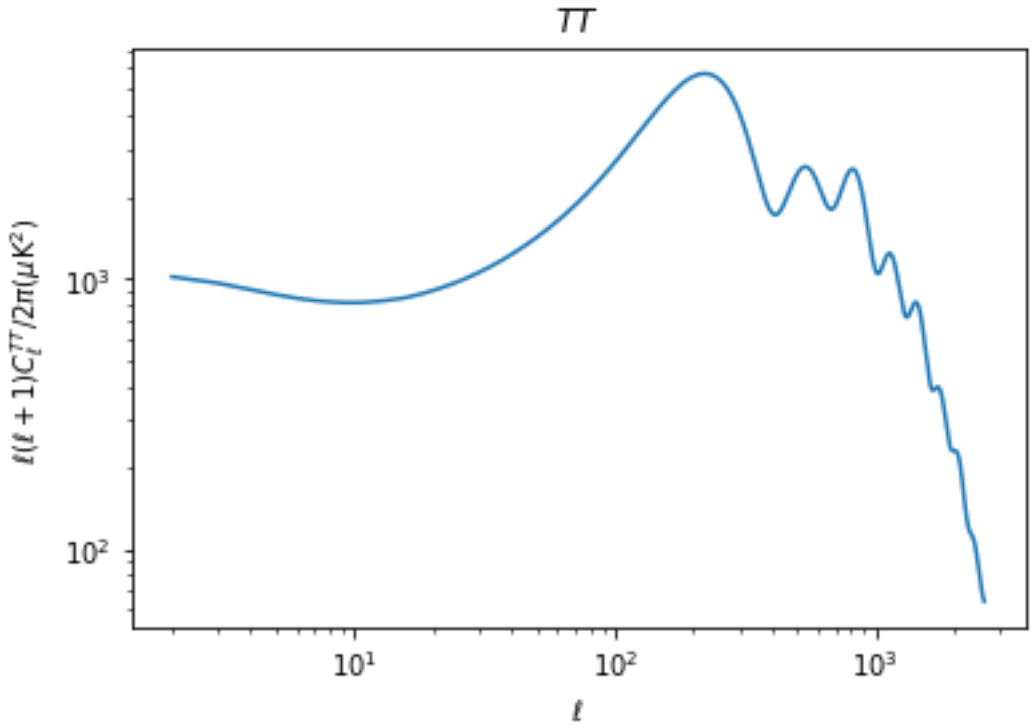

Figure 2.3: Angular power spectra of the CMB intensity calculated by CAMB using best-fit Planck data [1].

2.1.2 SZ simulation

We have simulated both the kinetic and thermal SZ effects. To do so we have used the Planck Sky Model[3,4] [7]. The Planck Sky Model (PSM) is a complete representation of the submillimiter sky, ranging from 1GHz to 1 THz. It summarizes, pretty well, the current knowledge of the GHz sky. It has a complete set of versatile programs and complete data that can be used in the simulation or the prediction of the sky radiation in the frequencies of the CMB experiments, particularly Planck.

We have generated three SZ all sky maps (same number as CMB maps) using the PSM. The simulation were carried out according to the Planck best fit cosmological parameters and astrophysical parameters. We simulated both SZ effects model based with relativistic corrections up to first order. Each one of the three generated sky maps have each of the nine Planck channels, altought we have worked with 5 (70 to

[3] https://apc.u-paris.fr/∼delabrou/PSM/psm.html
[4] http://pla.esac.esa.int/pla/#plaavi_psm

353 GHz) as it will be explained later. This maps, and as for the rest of the maps are given in units of temperature fluctuation (from the CMB temperature) in Kelvin, for the seven lowest frequencies. For the two highest frequencies, the units are usually expressed in MJy/sr which came more natural. To change from Jy/sr to thermodynamic temperature in Kelvin we have used the equivalency

$$K_{CMB}(K) = \frac{I_\nu}{(2k_b\nu^2/c^2 f(\nu))}, \tag{2.1}$$

with $f(\nu) = \frac{x^2 e^x}{(e^x-1)^2}$, where $x = h\nu/k_b T_0$, ν is the frequency observed, and T_0 the CMB temperature at $z = 0$ (today). This function is already implemented in the python library astropy[5], which we have used. Once we have all in units of K we change them to μK which are more natural. In figures 2.4 and 2.5 our simulations are represented.

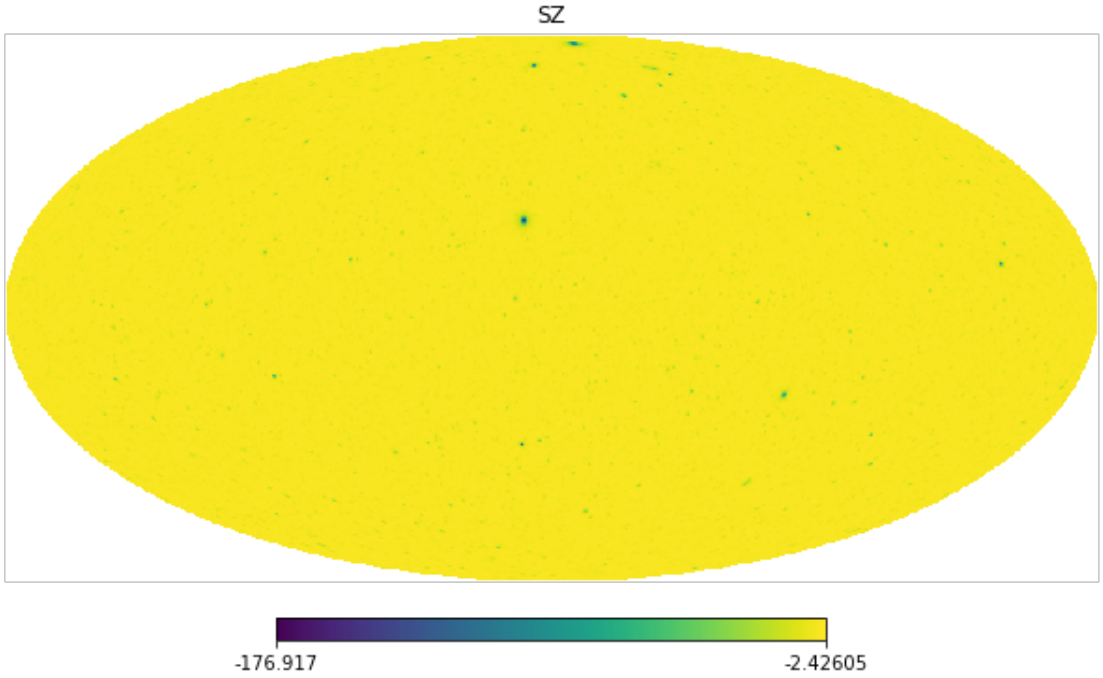

Figure 2.4: Projection if the all sky map simulated SZ foreground at 30 GHz. Units are given in temperature fluctuation in μK.

[5]https://www.astropy.org/

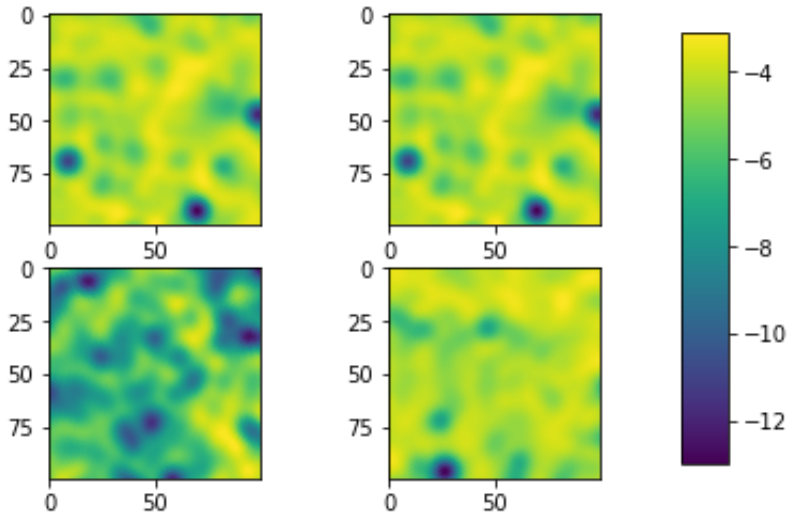

Figure 2.5: Gnomonic projection of the simulated SZ emission, at 30 GHz, at random coordinates. Units are given in temperature fluctuation in μK. The position of the clusters can be seen as dark spots.

These maps (the CMB and SZ maps) are just arrays of $5\sim\ 10^7$ positions (for each frequency), representing HEALPIX pixels, with a value. The addition of both maps came as easy as the sum of both arrays. We convolve the total map with a Gaussian kernel of the effective resolution, per channel, given in table 2.1. After all of this work a random Gaussian number with zero mean and standard deviation as the temperature noise level (see table 2.1, it is given scaled to $1^\circ$and we expressed it in given the area of our pixels). At this point we have our maps fully prepared, see figure 2.6.

Also for a test of our trained CNN, on CMB plus SZ and noise, on a more realistic map of the sky, we have added to this already built maps the other galactic foregrounds, simulated by the PSM, in the proper units and with a resolution equivalent to each Planck channel.

2.1.3 Patching of the sphere

There has been previous works, in the recent years, that have work on the implementation of convolutional neural networks on the sphere (see [24] and [25]). This are promising works, however they are not yet fully explored, while the use of CNNs has been well tested on Cartesian 2D images for a while, in fact, the use of CNNs in image recognition is one the most advanced areas of Deep Learning. Knowing this we have opted for the use of flat patches, of 100×100 pixels of $2.8^\circ\times2.8$, of the sphere as the input of our network. The use of flat 2D images is also the standard for the usual filter-based methods.

Other reasons to use flat patches as we have defined are: 1) The size of the pixel in the plane is similar to the size of the pixel in the sphere, allowing a good sampling of the maps at N_{side} = 2048, even though maps at lower N_{side} will have an over sampling. 2) 100×100 pixels in $2.8^\circ\times2.8^\circ$patches prevents distortions in the projection, this extension guarantees 1-2 more clusters per patch. 3) Smaller patches allows more patches from the same sky (this will be specially important when training a CNNs with foreground included), and it also reduced the memory needs.

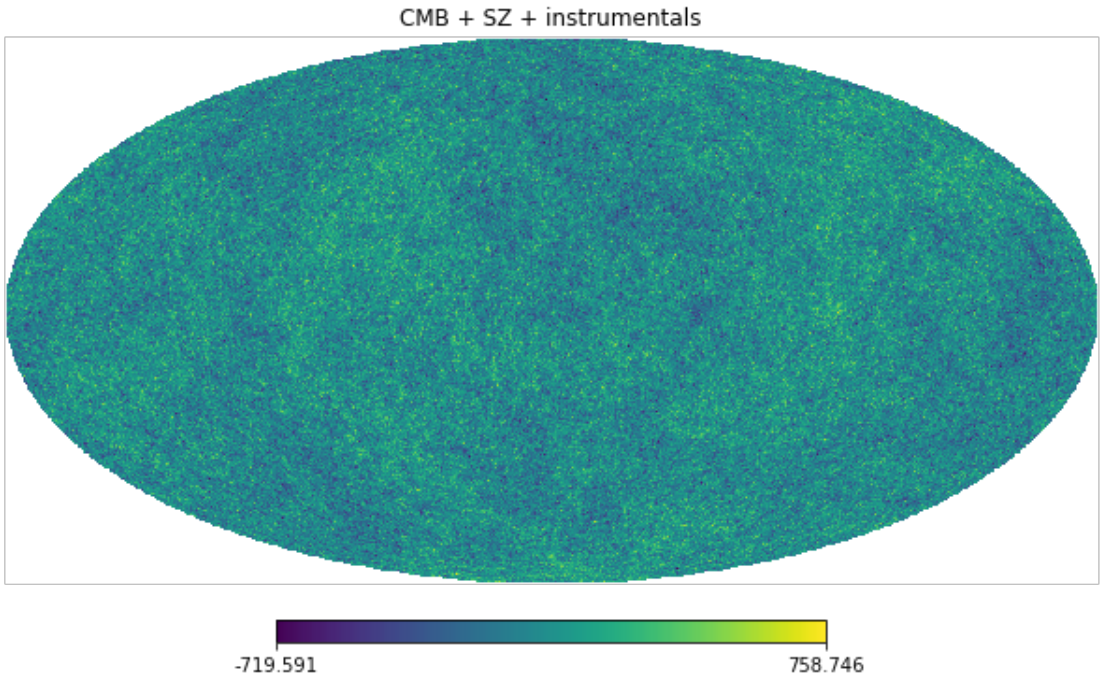

Figure 2.6: Projection of the all-sky we have used to detect the SZ effect. It contains the CMB, the SZ effects, Gaussian white noise, and it is smoothed to the corespondent Planck channel resolution of 70 GHz. Units are given in temperature fluctuation in μK.

The patching process is simple, we just used a gnomonic projection of the sphere in to a plane. A gnomonic projection consists in the projection, P, of a point in the sphere $P1$ from the center of the sphere O to the tangent plane to the center point of the patch S, this has been depicted in figure 2.7. To do so we have created, using healpy `gnomonicview`, patches of 100$\times$100 pixels covering an angular size of 2.8°$\times$2.8°.

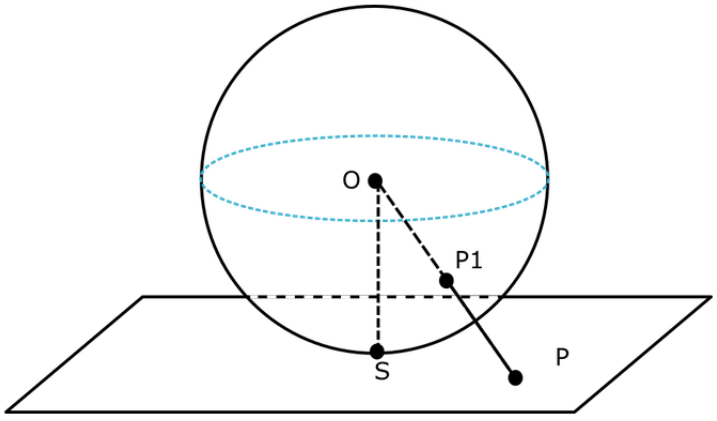

Figure 2.7: Visual representation of the gnomonic projection P, of a point $P1$ in the tangent plane at the point S. Source: http://blog.nitishmutha.com/equirectangular/360degree/2017/06/12/How-to-project-Equirectangular-image-to-rectilinear-view.html

This kind of projection creates a distortion, the image gets “stretched" at the borders (i.e., the pixels of the border of the image are more separated in the sphere than the pixels near the center). This effect is

important for very large patches, in our case, side=2.8°, this is not very critical.

Previous experiences with similar CNN applications, show that the overlapping of the patches in the use of this kind of networks tends to make worst predictions. The main reasons behind this are, that you can not guarantee the independence (not overlapping) of the validation and training sets, so the comparison of the loss and accuracy curves in the training will not illustrate the proper functioning of the network, and that the overlapping would make the CNN to work over the same areas, favoring the overfitting which makes the predictive skills of the CNN worse.

In order to evade this problem, we have opted for a patch distribution according to [26], this article includes several options for the partition of a circle into equal-are cells, and the equivalence in the hemisphere. We have imposed the condition of square, patches, providing the following distribution:

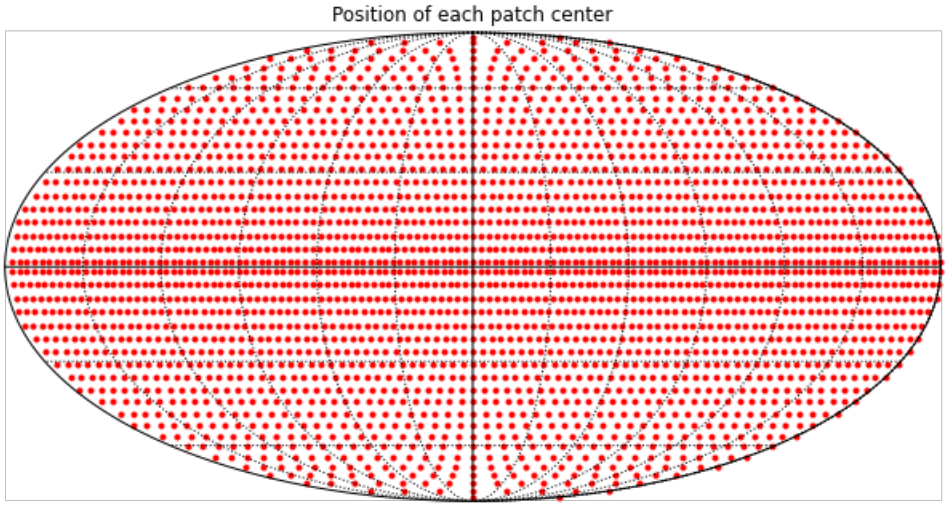

Figure 2.8: Position of the center of each patch in the sphere, according to [26], imposing the condition of square-not overlapping patches. Dashed lines represent iso-latitude (horizontal) and iso-longitude(vertical) lines.

This specific configuration gives us 2736 non overlapping patches, a randomly selected 80% of the patches were used for the training set and the remaining 20% for the validation set. Let us note that this number of patches, attending to their size, represent a 44.63% of the whole sky. At this point we have a set of 2736 arrays of shape 100×100 pixels in the .npy[6] format, which enables quick input/output operations, and a small file size. We need a conversion from a point (i,j) in the k image to a coordinate in the sky. This will be important later to represent the detected sources in the images at the sky. As we know the position of the center of the patch, and the pixel size, we can use this information to link a point (k,i,j) to a sky coordinate. In fact healpy has a class, **projector**, that among its functions have one that can link positions in the image to positions in the sky. We simply create the projector objects at the position of the k patch that we already know. Using this object functions we go from (i,j) in the images, to their correspondent coordinate in the celestial sphere.

[6]https://numpy.org/

Labels

We will perform a supervised training of our CNN, in order to do so, we need a set of labels (a solution to the clusters location in the images that the CNN uses to train) that will be given to the CNN with the CMB plus SZ data. From our maps of the SZ effect, we can detect the minimum values (or maximum in frequencies > 217 GHz) of the clusters emission. In this scenario each minimum is a confirmed cluster. The location of this minima can be studied on the sphere by a HEALPIX function, `hotspot`, it returns for all the minima, the pixel in the N_{side} of the map, and the coordinates in an specified coordinate system. With this information a label map was created for each sky, this maps consist in an array of zeros in each pixel position, except for those where the minima were located and also the neighbour pixels in a disk of radius 1.1 the FWHM resolution of the 353 GHz channel, give a shape to the position of the clusters in this map, helping the CNN to recognize the spatial distribution of the clusters. We have used the FWHM of the 353 GHz channel because it has the smaller resolution from the frequencies that we will work with, [70-353] GHz. We wanted a small FWHM to train a CNN able to identify with high precision the position of the clusters. This size for the label was chosen as a compromise, to do not have very extended objects to identify the SZ clusters. We will discuss later a bit more this. `hotspot` gave us the position of $\sim 5 \times 10^7$ clusters, homogeneously distributed, per map, following this distribution: This distribution will be of

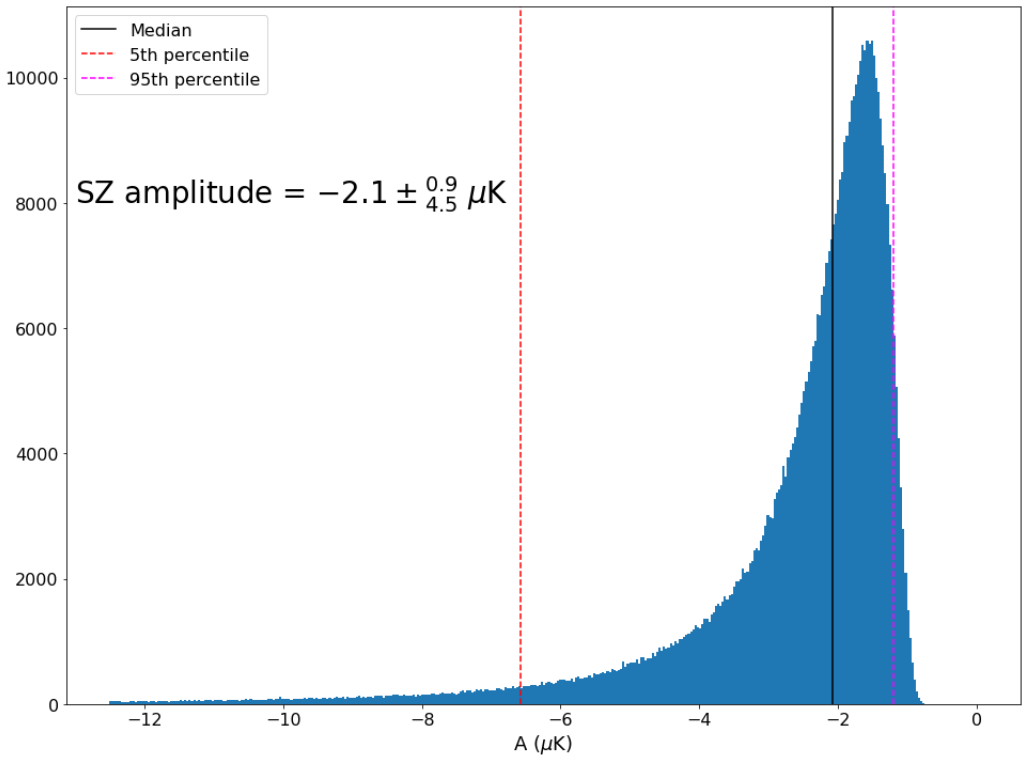

Figure 2.9: Histogram of the intensity of the SZ minima at the 143 GHz channel of the third sky realization. The median is given with the 5th and 95 the percentiles, covering a 90% of the population.

importance later. We selected the 2300 strongest sources as the labels, this is so because even if in this simple simulation we could easily detect many more sources, the next natural step will involve leading within that case, we would expect a cae similar to the PSZ2 [27] that have 1653 sources. In summary, to be able to calculate the efficacy of a future CNN with more realistic foregrounds we used a conservative number of detections as labels: 2300.

2.2 Convolutional Neural Networks

Convolutional Neural Networks [28] are a relatively new technique developed over the past two decades. They have reported huge advancements in the Artificial Intelligence field, becoming quite popular. It very is common to hear about new uses for CNNs. Physics is not a special field in this sense, as CNN applications are continuously appearing. We want to test the performance of CNNs on multi-frequency maps detecting the SZ effect. Our CNN has been built using the TensorFlow library [29].

2.2.1 Basics of neural networks

CNNs are a type of a neural network. Neural networks are described as mathematical functions that receive input data and produce outputs. Both the input and the outputs must be real tensors with fixed dimensions (an image for example). Classic neural networks handle 1D arrays, this limits its utility. CNNs, however, break this restriction, they can use 2D arrays or even higher dimension arrays such as an RGB image. A neural network might have different functions: there are models based on neural networks that can differentiate animals, or recognize objects/people etc..., there are others that can produce very realistic, but fake, news, other models can even produce art based on written descriptions, they can even be used as early illnesses detectors. The uses seem to be endless, and science is not excluded, machine learning is quickly becoming a must need tool in a scientist kit.

The basic unit of a network is the neuron. Neurons are usually grouped, although they can also be alone, forming in layers. We define the output of a given layer i as $x = (x_1, x_2, ..., x_s)$. This output is the input for the $i + 1$ layer, at this layer, a given neuron k will produce an output

$$y_k = f_i(x \cdot w_k + b_k), \tag{2.2}$$

where w_k is the weight vector, it indicates the importance of the outputs of neurons in the previous layers, b_k is a bias, and f_i is the activation function, it prevents the collapse of the layers in to single neurons, as the summed value of n neurons with no activation functions can be expressed as a single equivalent neuron. The typical neural network graph can be seen in figure 2.10. The most typical activation functions are:

- The **step** (thresholding) function. $f(x) = 0$ if x $<$ t, and $f(x) = 1$ otherwise; t $\in \mathbb{R}$.
- The Rectified Linear Unit (**ReLU**) function. $f(x) = \max\{0, x\}$.
- The **sigmoid** functions: bounded, differentiable, real function, defined for all real values, non-negative derivative at each point and exactly one inflection point. e.g., logistic function, $f(x) = 1/(1 + e^{-x})$.
- The **linear** function $f(x) = kx$, $k \in \mathbb{R}$.

What makes the neural networks special is that they can learn. Learning, in this context, means the network can change the weights of the neurons to achieve the optimal output. Neural networks are a

supervised learning algorithm, that given a pair of sets, data and labels, the network will adapt the weights in order to achieved a label-like result over the data training set. Neural networks do not need to know the reason behind the processes that make the inputs look the way they do, they just need a large enough data set and correctly labelled examples and it will find a solution for the problem given the nature of it, and a proper architecture. One drawback is that even if the model work, it is not trivial to get information of the characteristics of the inputs. Inversely one could realize an unsupervised training of a neural network, this will help retrieving information from our data.

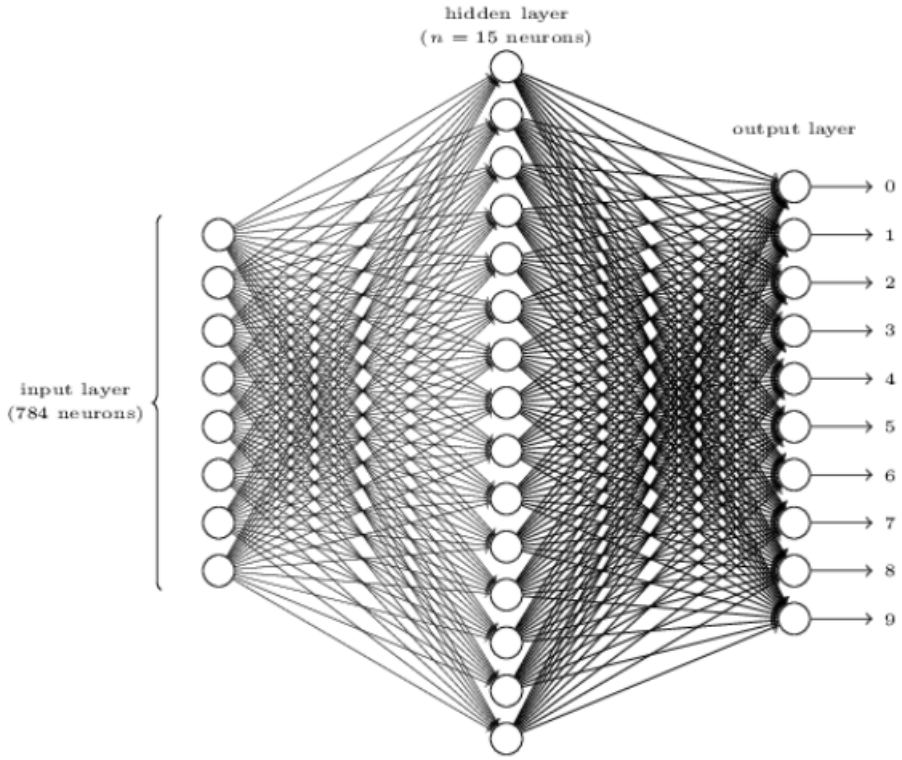

Figure 2.10: Example of a neural network structure. Source [30].

Training: loss, accuracy and predictions

As previously explain a neural network is an algorithm or function that given an input produces an output. This is called the prediction of the network. This prediction depends on the architecture of the network and on the weights. The architecture defines the functionality of the network while the weights adjust the network to the data of our problems. There are hundreds if not thousands of parameters in neural networks. The optimization of their values is what we want from the training process. This process is the time in which the weights are constantly changing to obtain better predictions. To explain how this weights change we must introduce the loss function.

The loss or costs function is an an optimization algorithm, it is the function used to evaluate a candidate solution, i.e., a set of weights. The loss function must be carefully taken. It generally depends on the purpose of our network. There are roughly four different general purposes:

- **Regression:** Inferring of a continuous variable. Number of persons in an image.

- **Classification:** Determine if the input belongs or not to a class. Is there a person in an image?
- **Multi-class classification:** Similar to the classification but with a higher number of classes. Is there a baby in the image?A teenager? etc...
- **Segmentation:** Point wise classification, i.e. determining areas that fulfil a condition. Where is the person?

Our supervised method requires two inputs for the training. A set of pairs, training data and the correct answer to the problem, the label. In our case our CNN will be a binary segmentation: The network must predict, at each pixel, if the image contains a cluster, and how likely. We will use then the binary cross entropy as the loss function, the most common in this kind of problems.

The binary cross entropy is (see [30] for a detailed explanation):

$$H = -\frac{1}{r}\sum_{j}[y_j \ln O_j + (1 - y_j)\ln(1 - O_j)], \tag{2.3}$$

where each y is the value of the label at the output neurons and the Os are the predicted values.

There is also an important function called the accuracy function. The accuracy function is a parameter that gives and indication of how the network will work with external data. Another set of data and labels is given to the network at the time of training, however this data is no used for the weight estimation. Different loss functions have been used in this kind of problems with poorer results [31]. For the accuracy

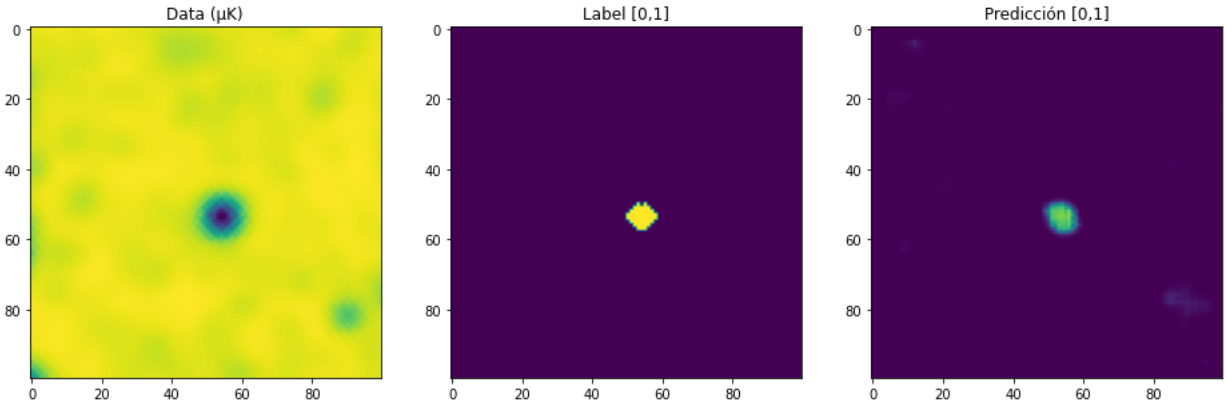

Figure 2.11: Example given of a train (left) - label (center) pairs, with the prediction of our CNN (right). In this case the left image is just the SZ, our data train also includes the CMB and noise, this is would be a fictional train-label pair for an easier understatement.

function we will use the binary accuracy, a ratio of correctly classified pixels given a threshold. This parameters determined the training and also helps to state some conclusion of the overall performance of the network over the training and validation data sets.

The training begins with the division in batches of the training set.The totality of the set is not evaluated at the same time, but batch by batch, which reduces the computational time and the memory

needed. The batches perform their predictions, the loss function is calculated, the network parameters are recalculated to look more similar to those that obtained the minimum loss function. This recalculation is done by a gradient descent of the function with respect to the weights. We selected the Adam optimiser [32]. The training tries to minimize the loss function, while the accuracy is the metric that indicates the performance of the network, but it has no implication with the weights recalculation.

The number of epochs this training last is an hyper-parameter of the CNN. At the end of each epoch the performance is tested, the weight are fixed and a prediction is made on the validation set, which is as the training set but smaller in size (we have used an 80-20% of the patches for training-validation). It is important to comment that the validation pairs have not been used in the training. This is a common Machine Learning technique. The predictions are compared to the validation labels, and the loss/accuracy functions are computed again. The validation performance will always be a bit worse that the training performance, however the validation provides a more realistic estimation of the performance in a real scenario with unlabeled data. If the difference between the validation and the training loss/accuracy functions increases to much, the model is overfitting, it has recognized the local patterns in the training data, this local patterns are of no use in external data, this should be avoid, stopping the training earlier, or changing the architecture of the network. To identify overfitting, one can look if the validation loss stops decreasing at some point while the training loss keeps decreasing (or inversely with accuracy), see figure 2.12 for an example.

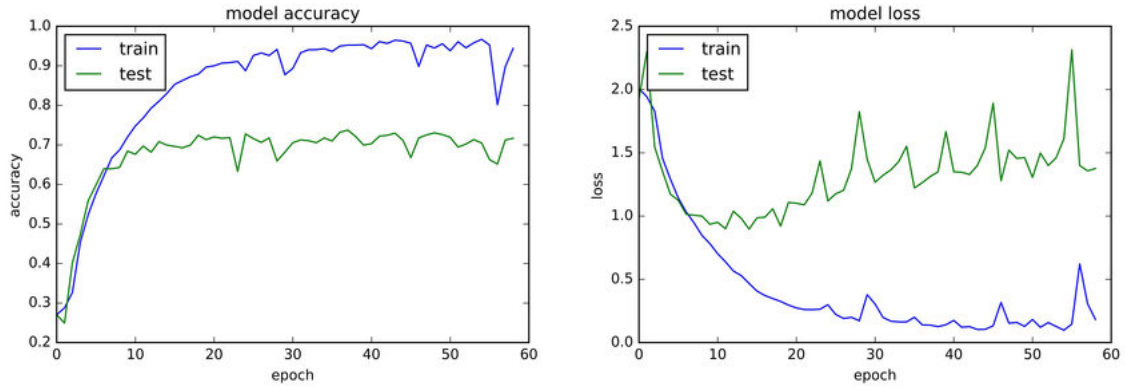

Figure 2.12: Validation and training loss (right) vs accuracy (left) in an overfitting case. Source: [33].

Convolution layers

CNNs are a special type of neural network that have the convolutional layers that gave them their name. This layers are special because they can handle inputs with higher dimensions, not just 1D. This layers are a composite of kernels (filters), of dimension m_x and m_y, always smaller than the input array dimension. The result of each convolution is known as a channel, this layers also have activation functions. A generalization of equation 2.2:

$$F^r_{a.b} = f\left(\sum_{i=0}^{m_x}\sum_{j=0}^{m_y}(W_r)_{i,j} \times I_{a\times s_x-(m_x-1)/2+i,b\cdot s_y-(m_y-1)/2+j}\right), \tag{2.4}$$

where I is the input array of dimensions n_x, n_y, and W is now a weight tensor. (In this case it was assumed that bot m_x and m_y were odd integers, to make the expression simpler). The new dimension of

the convolved input is $\frac{n_x}{s_x} \times \frac{n_x}{s_y}$ ($\times$ k, the number of filters). s_x and s_y are the strides, the spatial step in the x and y direction between each convolution, see figure 2.13, for a visualization of a filter convolution.

It may happen that if our kernels have a dimensionality > 1, some of the indices of the filter during the convolution, might be outside the input space. The padding, adds zeros at those positions allowing the convolution of the hole input tensor, maintaining the dimensionality. No padding will mean that the edge values will not be fully used, losing dimensionality.

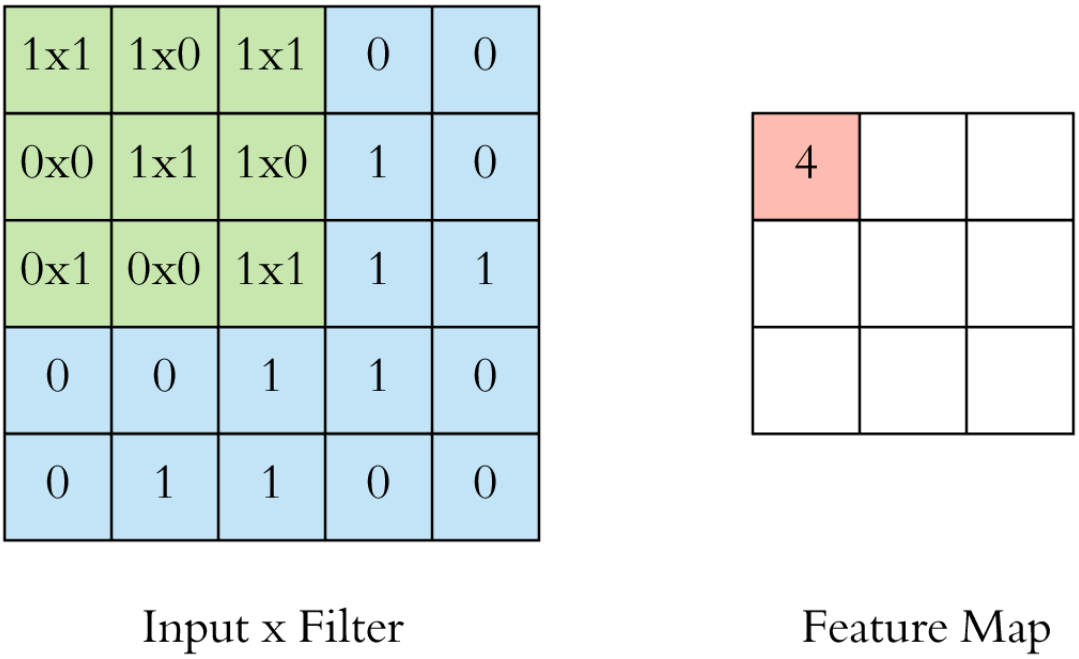

Figure 2.13: Representation of a 3×3 filter convolution. Strides: $s_x = s_y = 3$. Input shape: 5×5. No padding applied, 3×3 output shape. Source: https://morioh.com/p/f99fa6bd2337.

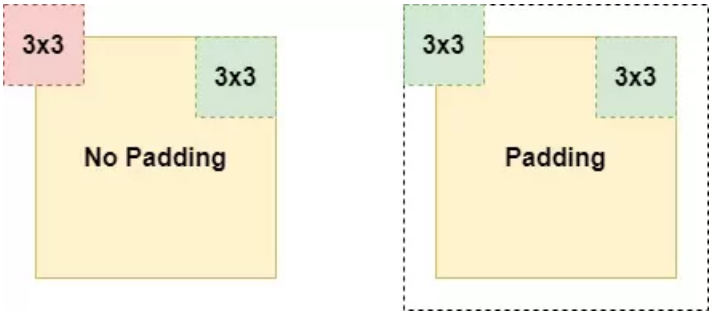

Figure 2.14: Example of the dimension reduction of the output if no padding is used. (Left) no padding, values at the edge can not be used in the convolution. (Right) padding is allowed, using every values of the input. Source: https://www.machinecurve.com/index.php/2020/02/07/what-is-padding-in-a-neural-network/.

2.2.2 Network architecture

The way a network works strongly depends on its architecture. First the input layer must handle the shape of the input data, then the output layer must be adequate for the purpose of the model, a binary

segmentation in this case. The output layer must consist in a single filter (each pixel acts as a neuron), its activation function must works values in the range [0,1].

There are other important layer used in our CNN apart from the convolutional layers:

- **Max Pooling:** It divides the input into sub arrays, and reduces the dimensionality by only keeping the higher value. Pooling layers reduce the data size, thus decreasing the number of weights. This

$$\left(\begin{array}{cc:cc} 3 & 5 & -2 & 1 \\ 4 & 0 & 0 & 6 \\ \hdashline -2 & -4 & 1 & 0 \\ 1 & 0 & -1 & 2 \end{array}\right) \xrightarrow{\text{Max Pool}} \begin{pmatrix} 5 & 6 \\ 1 & 2 \end{pmatrix}$$

Figure 2.15: Max pooling example: The original 4×4 matrix is divided into 4 2 × 2 submatrices, from each submatrix only the higher value is conserved and new matrix is formed.

  prevents overfitting and accelerates the convergence, they might even help detecting features of the data. However it must be done carefully, as a extensive use will produce worse predictions, accuracy is lost in the identification of the cluster position.

- **Group normalisation:** It organizes the channels into different groups, and computes the mean and the variance and along the height and width axes, and along a group of channels. The inputs are normalized for each group according to their mean and variance. This is done to avoid numerical instabilities, allowing higher learning rates.

- **Transpose convolution:** It transforms the input in the opposite direction of a normal convolution. We use this to increase the dimensionality of the data. This is important after max poolings or convolutions with strides that reduce the dimensionality (figure 2.13).

The design of a network is not an easy task, the variability in the number of layers, its depth, the type of layers, and the activating functions, makes almost impossible to find the optimal architecture, there might always be a better one. The previous works realized at IFCA [34], and studies such as [35] and [36], have gave us experimentally validated guidelines for the network architecture:

- Use raw arrays of data as inputs. If we do not normalize the data the network would focus on the shape and not the brightness.

- Use only convolutional networks. Our aim is to detect localized and small features. Introducing dense layers can decrease the predictions of the cluster positions.

- Normalized layers would help working with the large values of the arrays, also the training will be faster.

- Using a sigmoid activation function, our predictions will be given in values in the range [0,1], this is a kind of “probability" given by the network.

Given the quickly evolution of this field, there might be better architectures, that we do not know about. The decisions taken are based on the experience of the group, however, testing new configurations would always be a good idea, CNNs have not being around for a long time, so there is much to discover.

The architecture of our CNN is as follows. The first block receives a 5 100×100 images, one per frequency, that is maintained through the block, the first layer is a 2D convolution layer of 16 kernels of size 7×7 and strides $s_x = s_y = 1$. It is follow by a group normalization layer (16 groups) with the same amount of strides, a ReLU activation layer, now the same scheme is repeated for the second block but the convolutional layer has the double of kernels, and the group normalization has the double of groups (32), the dimensionality is conserved. The third block introduces a max pooling that reduces the dimensionality the 50×50, its size is 2×2, and the strides are $s_x = 2, s_y = 2$. The activation function is also a ReLU. The last block is a transpose convolution block, it has a 2d transpose convolution of 64 kernels, of size 5×5, and strides $s_x = 2, s_y = 2$, to double the size of the image up to 100×100 once more. It follows by a group normalization layer of 64 groups, and a ReLU function. It ends with a sigmoid activation function that gives an output between [0-1]. In total this particular CNN has 80,609 parameters. The summary of the architecture is fully given in table 2.2.

Nº	Layer type	k	Input size	Output size	Filter size	s_x	s_y
1	2D Convolution	16	100×100	100×100	7×7	1	1
2	Group normalization	-	100×100	100×100			
3	ReLU	-	100×100	100×100			
4	2D Convolution	32	100×100	100×100	7×7	1	1
5	Group normalization	-	100×100	100×100			
6	ReLU	-	100×100	100×100			
7	Max pooling	-	50×50	50×50	2×2	2	2
8	ReLU	-	50×50	50×50			
9	2D Transpose conv.	64	100×100	100×100	5×5	2	2
10	Group normalization	-	100×100	100×100			
11	ReLU	-	100×100	100×100			
12	Logistic	-	100×100	100×100			

Table 2.2: Architcture of the used CNN for the detection of SZ emision in CMB + noise maps. Last two columns represent the strides, k is the number of filters.

2.3 Object extraction

This section explains the detection criteria used on the predictions made by the CNN and how do we evaluate the CNN from this predictions.

2.3.1 Detection criteria

Once we have a trained CNN, we feed it with new data, different from the validation and training sets. The CNN returns images of the same shape as the inputs, each pixel have a value between 0 and 1, shaping the position of each cluster as predicted by the network. To consider a detection we need:

- A group of bright pixels (values > 1) clearly resolved and surrounded by black pixels (value ~ 0).
- The group must be large enough (have enough pixels). It requieres a minimum size.
- The edges must be taken carefully there might be convolution effects.

For this we have used a routine from the scikit-image[7] package, designed for the scientific analysis of images. The so mentioned function is `peak_local_max`, and finds peaks in an image, returning their (x, y) indices. A peaks is a local maxima in region of 2 * *min_distance* + 1 (min_distance is 1 by default, but set it at 5 to avoid double detections on extend predictions), this function requires a *threshold* value to consider the maxima as a peak. The function can ignore the peaks near to the edges, although we finally took this peaks into consideration as the network did not produce any strange effect on the edges, at least not important distortions were noted.

2.3.2 Network evaluation

Once we have our network trained, and applied to data (different from the training/validation data), getting predictions. When we have the use the detection criteria, given a detection threshold, on this predictions and we have localized the positions of the predicted clusters, we need to determine the good predictions, the spurious predictions, and the missed clusters. We will evaluate our CNN based on the total number of predictions, the reliability R and the completeness C, this two parameters both depends on the number spurious detections, the number of cluster not predicted, and the number of confirmed predictions.

Our evaluation algorithm should do the following:

1. Return the number of detections.
2. Count the number of detections and the number of clusters in the SZ map above a certain T_{cut}.
3. Count the numbers of detections that do not correspond to an actual source (cluster), these are called spurious detections.
4. Count the detections that actually corresponds to a confirmed source, these are called real detections.

Our algorithm calculates the peaks fulfilling the explained criteria, for a given cut in the networks output. The position indices of each peak, as well as an identification index of the patch are stored in two different

[7] https://scikit-image.org/

arrays. If the location of the peak is located within a bright spot in our image containing the labels (the value of the pixel is 1), or is at least at a distance of 3 pixels (2.8°/ 100 * 3 = 5.04 arcmin), the peak parameters and the value of the source at that point, in the data patch, is stored in the real detections array, if not in the spurious.

We have know the number of real detections, spurious detections, and total detections. The reliability R, and completeness C.

$$R = \frac{\sum real\_detections}{\sum(spurious\_detections + real\_detections)}, \tag{2.5}$$

$$C = \frac{\sum detected}{\sum(detected + undetected)}, \tag{2.6}$$

where the completeness is defined over a total of sources above the source of the fainter detected source.

Chapter 3

Detection in CMB with SZ and noise

3.1 Method characteristics and evaluation

The CNN has been trained with two different skies (44.63% coverage, each), $\sim$ 5500 patches (80% for training, 20% validation), for 10 epochs, a batch size of 10, and a learning rate of 0.01. As mentioned earlier we have decided to use only five channels, from 70 GHz to 357 GHz. With this channels we can still observed the SZ effect dependence with the frequency (negative for frequencies below 218 GHz, and positive for higher frequencies), this is to reduce the memory requirements of the method and to reduce the training times.

All the CNN related computations were made at the computer cluster Altamira[1] as the technical requirements were to high for the use of a personal computer. The training took $\sim$ 1.5h, approximately 10 min/epoch, using a node of 16 CPUs and a memory of 64 GB. In total 80 609 parameters were fitted. The validation and loss functions per epoch behaves nicely, however this plots can only tell us that there is no apparent overfitting, and that the CNN has been correctly trained, but we have no indication of how good can the CNN be.

From image 3.1 we have a first idea of the goodness of the network. There is nor overfitting, the validation and training loss and accuracy remains In figure 3.2 shows an example of the SZ emission at different channels can be observed. In the CMB + SZ + noise patch, the source is no longer visible, however in the CNN is able to fully determine the position and the size of the cluster, in this case. Before using our detecting criteria algorithm

To study the CNN we have applied the trained model to the third simulation of the sky. We also have a map containing the labels for this sky realization. The idea here is to search for maxima in the prediction patches given by the CNN. We first look at some random patches, comparing the prediction, the labels, and the actual SZ effect (figure 3.3).

A quick view on figure 3.3 can gave us some hints, in order to fully understand the CNN, and

[1] https://confluence.ifca.es/display/IC/High+Performance+Computing

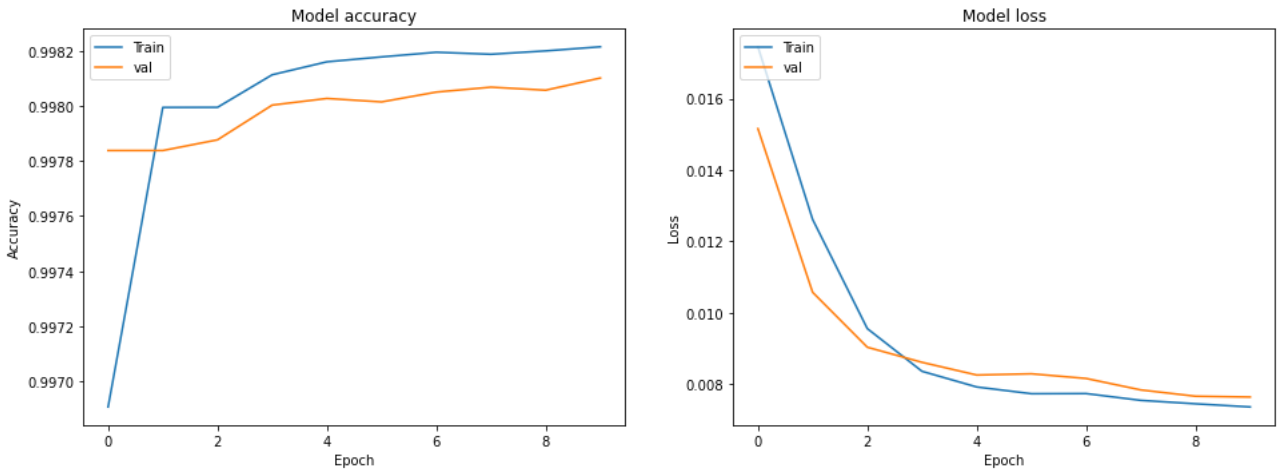

Figure 3.1: Training and validation accuracy (left), and loss (right) functions, per epoch, for the trained model.

characterizing its efficacy: 1) there are predictions made by the CNN where we do not have a matching label, this might be an spurious detection or a real detection of source not included in the 2300 strongest sources, 2) both the labels of the first and second patch correspond to a cluster of the same intensity ($\sim -24\mu$K), however the prediction of the label in the second patch is predicted with a value about half the value of the first. This could be caused by the influence of a third label pretty closed to the second (center patch), or maybe there is no relation between the temperature of the cluster and the output of the CNN, this would be a result worth the study.

We finally measure the number of detections, based on the threshold of the CNN output: If we set a threshold of 0.5, as a first approximation we observe obtain the following:

- The CNN predicts a total of 858 sources.
- 80 sources are confirmed as spurious (9.32%).
- The completeness is of a 22.32%.

This histograms contain pretty useful information: 1) The spurious detections have the same sign as the expected SZ emission in each frequency. If the spurious detections were to be linked to a CMB primary anisotropy we would not observe this perfect sign fit to the SZ emission. This suggests that either the CNN have learned from the SZ frequency dependence and only mistakes sources that follow a similar dependence, o 2) Many of the confirmed spurious detections are in fact real clusters that have not a label as they are weaker than the 2300 strongest sources, suggesting that we have a comparison catalogue too small. The fact that the CNN can detect a source weaker than the sources given as labels, is in principle a good thing, the CNN can detect weaker sources, but as we have some not a good completeness this suggests that the CNN output and the source have not a clear relationship, this will be studied later.

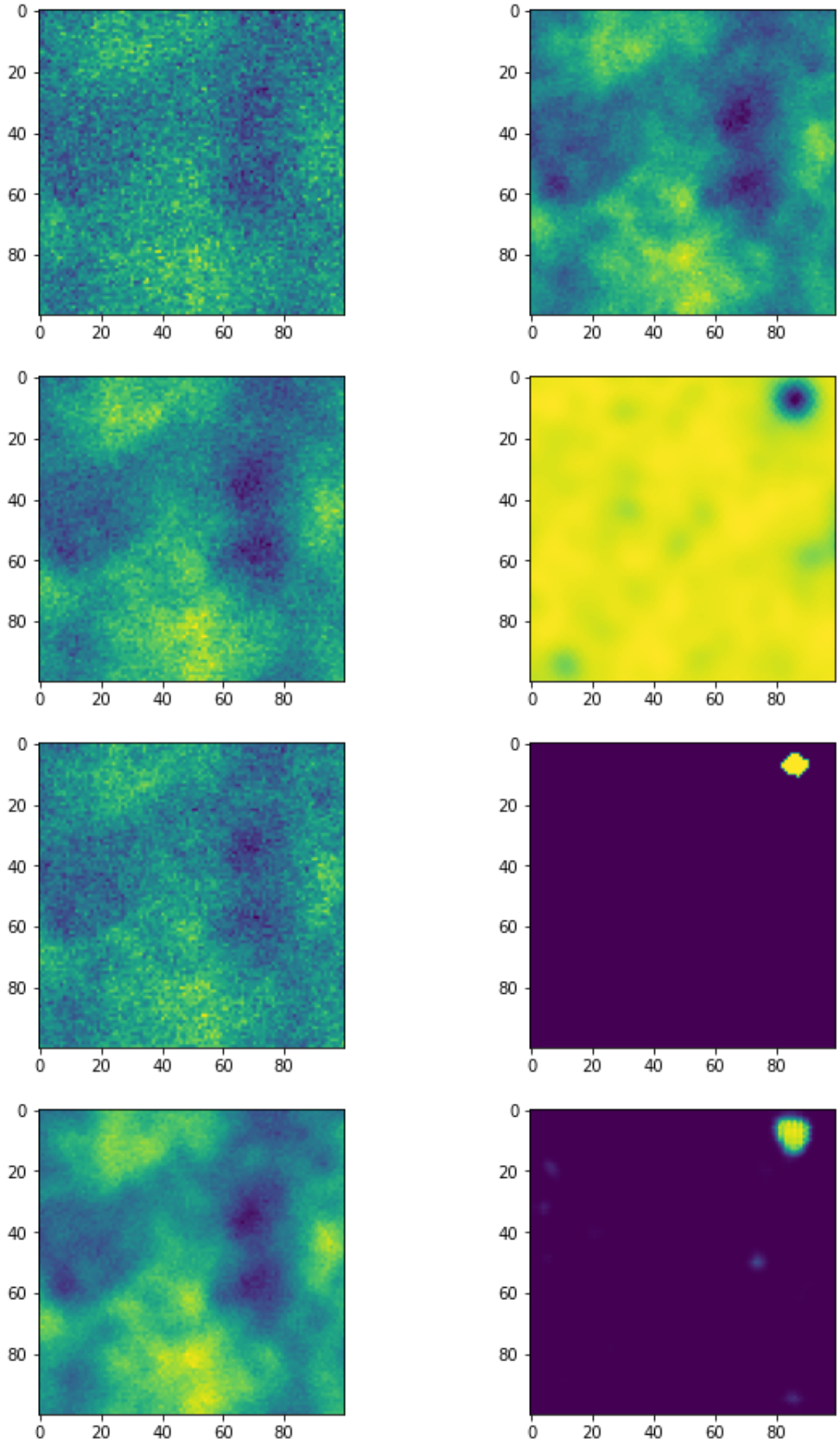

Figure 3.2: Representation of the same patch of the sky for different realizations. From top left to bottom left right: CMB + SZ + noise emission at 70, 100, 143, and 217 GHz. From top right to bottom right: CMB + SZ + noise emission at, SZ emission at 70 GHz, label, CNN prediction.

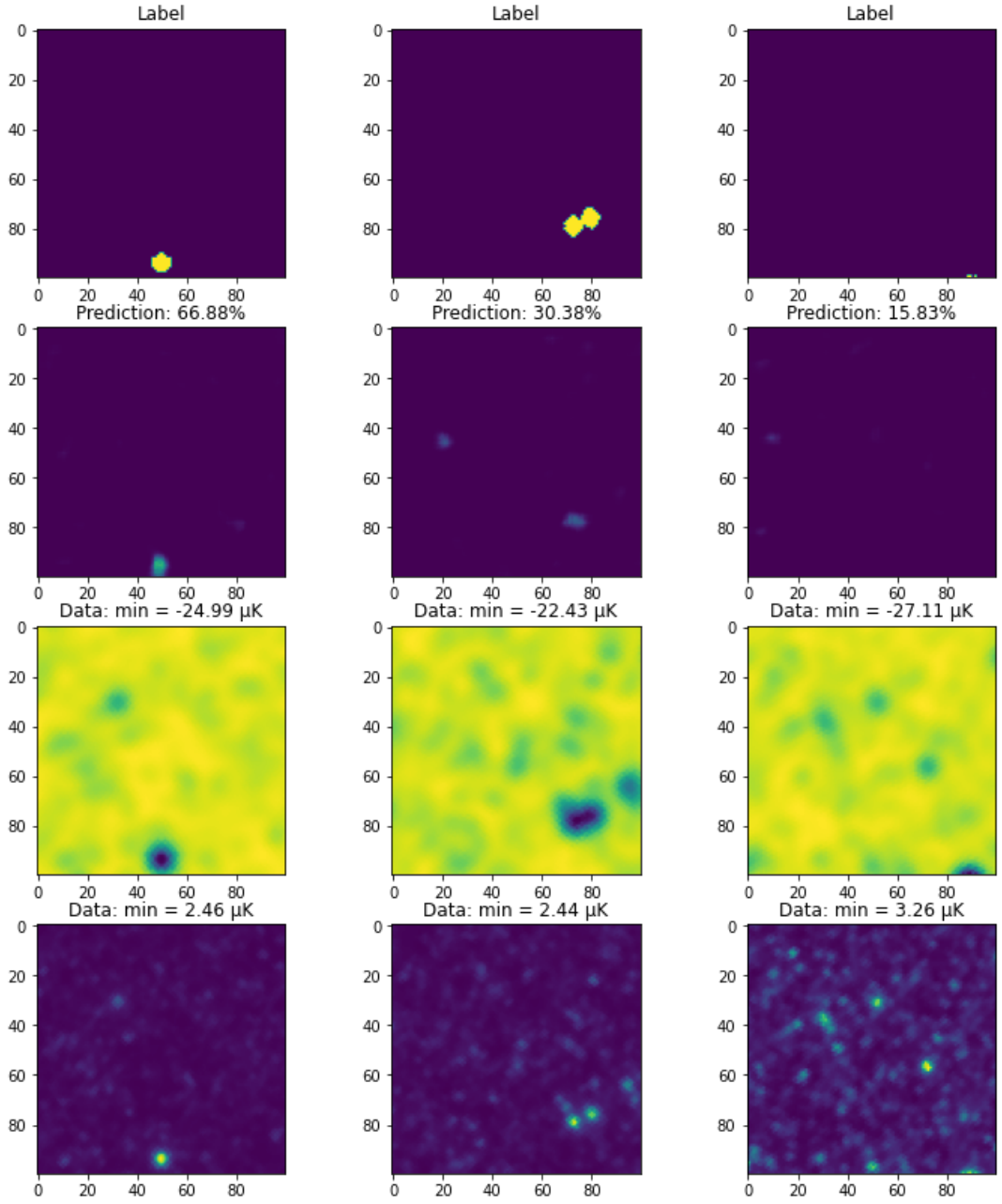

Figure 3.3: From left to right, different random patched of the sky. From top to bottom, label, CNN prediction, SZ emission at 70 GHz, and SZ emission at 353 GHz. Prediction indicates the max value given by the CNN. Data min indicates the coldest spot of the image due to the main cluster.

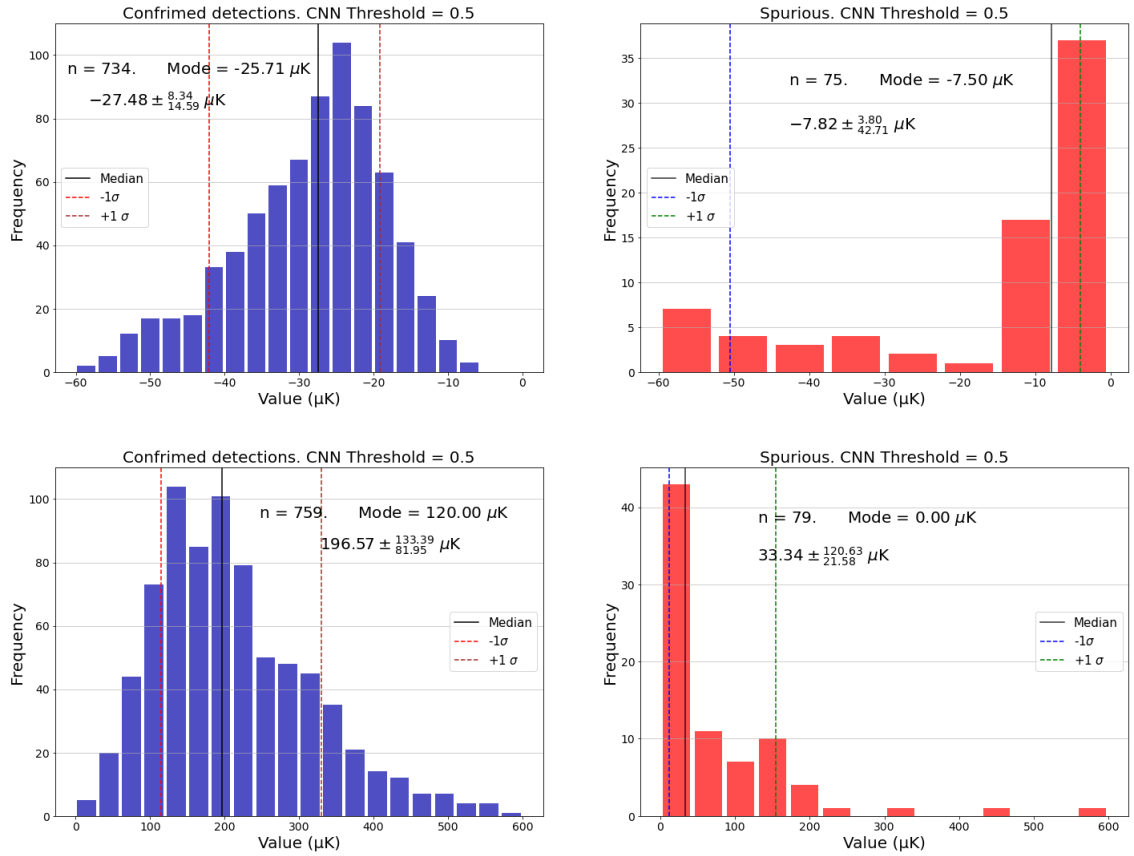

Figure 3.4: Histograms of the confirmed detections (left) and spurious detection (right), for the 70 GHz map (top) and the 353 GHz map (bottom).

We have tried comparing our results for the same threshold of the CNN output (i.e., the same candidate sources), with a depper catalogue of the strongest 50 000 sources, denominated 50k catalogued, here are the results:

- The CNN predicts a total of 858 sources.
- 54 sources are confirmed as spurious (6.76%).
- The completeness is of a 11.98%.

It seems clear that our first catalogue of 2300 sources, even if reasonable for a more complex simulation of the all-sky components, is a naive approach to study our CNN. The next step is to calculate amount of source predicted at different thresholds and the spurious detections reduction with the depth of the comparing catalogue.

A proper analysis of the how the miss-classification of correct detections into spurious predictions, is partially solved, is given in table 3.1. The stop at the 250k catalogue is not an arbitrary selection. Even though it is plausible that the intrinsic number of spurious detections might be smaller for thresholds of 0.3 or lower, however we have decided not to include depper catalogues up to the maximum which

would include $\sim 5.4 \times 10^5$ sources, as our determination criteria would no work properly: One of the reasons of using 2300 clusters in the initial catalogue is that with this number we will normally have one cluster/labelled per patch: this is essential to the CNN to have a proper training, but it also helps in the determination of matches.

Thresholds:	**0.05**	**0.10**	**0.20**	**0.30**	**0.40**	**0.50**	**0.60**
N**:**	6461	3607	1977	1362	1053	858	707
Catalogue depth	N_{spur} **(%)**						
10k	4358 (67.45%)	1899 (52.65%)	672 (33.99%)	296 (21.73%)	147 (13.96%)	80 (9.33%)	43 (6.08%)
50k	3292 (50.95%)	1400 (38.81%)	481 (24.33%)	201 (14.76%)	101 (9.59%)	58 (6.76%)	32 (4.53%)
100k	2673 (41.37%)	1132 (31.38%)	402 (20.33%)	172 (12.63%)	91 (8.64%)	54 (6.29%)	31 (4.38%)
150k	2204 (34.11%)	923 (25.59%)	331 (16.74%)	143 (10.50%)	74 (7.03%)	44 (5.13%)	30 (4.24%)
200k	1864 (28.85%)	728 (21.28%)	293 (14.82%)	129 (9.47%)	70 (6.65%)	41 (4.78%)	30 (4.24%)
250k	1581 (24.47%)	669 (18.55%)	248 (12.54%)	105 (7.71%)	63 (5.98%)	38 (4.43%)	29 (4.10%)

Table 3.1: Generalization of the CNN performance: The first and second rows are the threshold to consider a maximum in the CNN output image, and the amount of predictions (integrated number of peaks above this threshold for all patches). From the third row on, each row indicates the number of confirmed clusters above the whole sky (we are using half of it), that are being used to study the reliability of the predictions. 10k means 10 000 sources, and so on. Each element for a catalogue and a threshold represents the number of spurious detentions using our algorithm (detection criteria previously explained see chapter 2, section 2.3.1). Extrapolated to a whole sky N and N_{spur} would be approximately $2N$ and $2N_{spur}$, but the percentage of spurious sources, thus, the reliability would be the same). Green values are those at which N_{spur} saturates, i.e., getting into deeper catalogues does not improve significantly the reliability, the remaining spurious predictions are probably just bad predictions of the CNN, while in the other cases a given percentage can still be good detentions that need a comparison to a better catalogue.

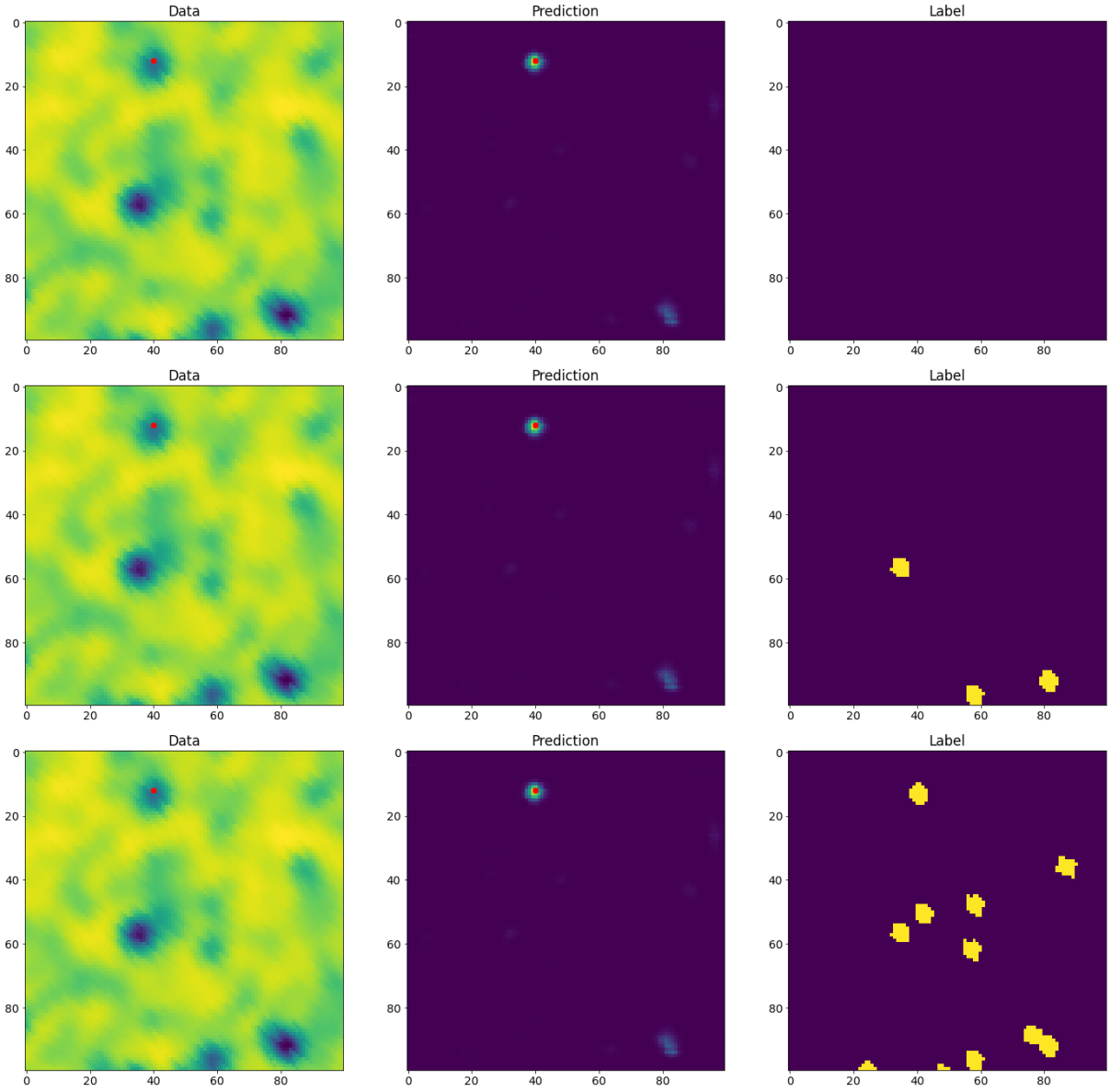

Figure 3.5: Determination of a correct match by comparing with deeper catalogues.2.3k (top), 10k (middle), 50k (bottom).

The problem why using catalogues too large can be guessed from figure 3.1. The deeper the catalogue the higher the number of labels per patch, if the number is too big, they might overlap and the real intrinsic spurious sources would be false positives.

We have already stated that is seems like the flux of the sources, and the CNN outputs are not related by a simple law. To study this we will plot the temperature of the confirmed clusters, vs the prediction of the CNN for this cluster. From figure 3.6 we can observe that a great portion of the points corresponds to low-bright cluster with a low CNN prediction counterpart, which makes sense if we assume the CNN prediction as a probability, however we also observe many low-bright clusters with a huge CNN prediction. Finally also many relatively bright clusters, and the majority of the extremely bright clusters have poorer predictions.

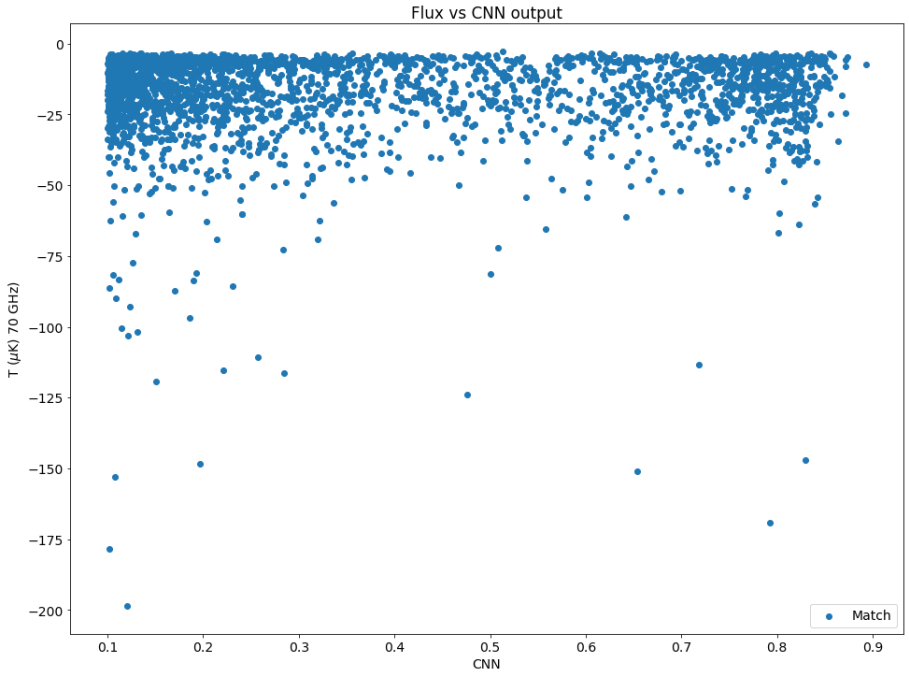

Figure 3.6: Temperature of the clusters in the 70 GHz map vs the prediction of the CNN for those clusters.

3.2 Discussion

There are several points worth mentioning in this section. We will start with a discussion on the performance of the network, followed by a the problem of the fully characterization of the network, and the prediction-flux problem, ending with a summary with some conclusions.

The first thing that we noticed is that the network works, it is capable to estimate the position of clusters through the SZ emission. The best case that we have been able to study is the threshold of 0.4 tested on the 250k catalogue. In this method we find 1053 sources, this can be extrapolated to 2000 sources, with a reliability of 94.02%, only 63 spurious detections per 1053 predictions. There is a recurrent

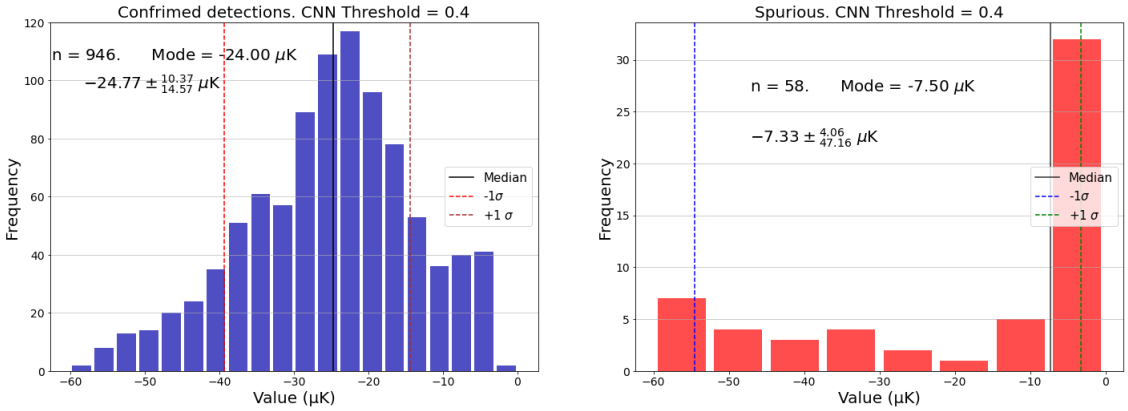

Figure 3.7: Histograms of both correctly detected (left) and spurious predictions (right), for the best set of catalogue generated.

problem here that is repeated at each cut. There is no correlation between flux and the CNN output. This

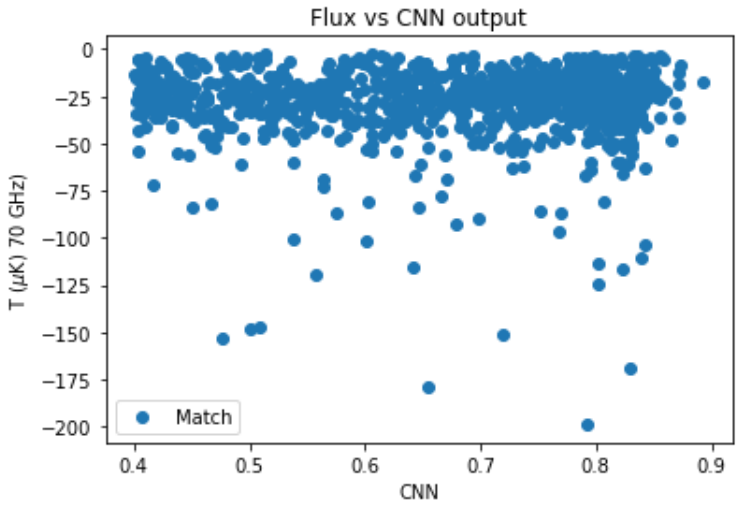

Figure 3.8: CNN output to cluster source, at the 0.4 cut.

makes that if we define a set of labels as the 100th or 250000th strongest clusters, to test the predictions, as the predictions treats equally bright and weak sources, we need to include a huge number of sources in this way to have these relatively low sources that before where taken as spurious. This can be done until a given point, as it was shown in table 3.1. If we introduce many labels into our images the spurious sources will be mistaken as good predictions. In this context the CNN works, but not as good as expected, we would want more detections bu the underlying problem here, is that the CNN does not associate the brightness of the cluster to the probability. In fact for each cut we have a distribution of low flux clusters and a few bright objects, for example in the 0.4 cut: We might think that this problems comes from the definition of our labels. We only included the 2300 strongest sources, but each source is represented as the same label, a fixed disk of a given size than have only 1 as a value. With this we have seen that our CNN have been able to learn about the size of the clusters, that is why it predicts clusters with different brightness with the sames probabilities, or even the weakest with the higher probability.

A possible solution would be the implementation variable labels. This would work by changing the size of the label according to the brightness, as the strongest cluster appear bigger this type of label might help working in this direction. Also, instead of using only the most strongest clusters as label candidate, which makes sense in realistic conditions where they are the ones that could be possibly measured., we can given labels with this variable size, and select them randomly.

We have to mention that as we have to include larger and larger data sets of labels to fully characterized the predictions, the amount of missed labels creates pretty bad completeness, as we saw with the 50k catalogue, we detected 858 sources over thousands of clusters (the completeness is not N/50k as some of those 50k falls in the other half of the sky we did not use). A completeness analysis is not possible in this case.

Even if we did not have not as much sources as we could expect, we have been able to use multi-frequency maps, this is an remarkable goal of this work, the use of multi-frequency to study the SZ effect

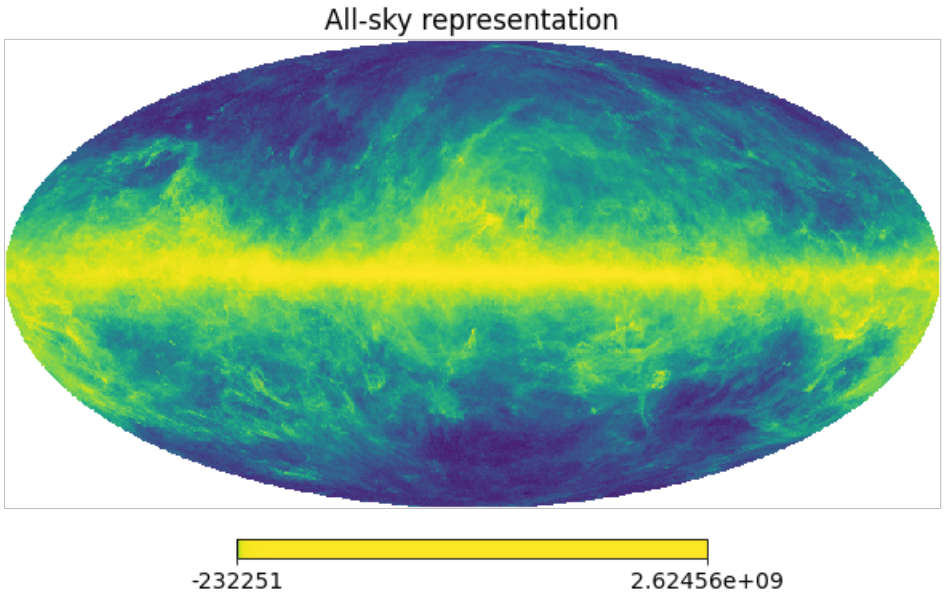

Figure 3.9: Representation of the all-asky simulation at 857 GHz, in μK.

is a promising indication. We have been able to determine the first difficulties on this line.

3.3 Detection in realistic scenarios

We ended last section with a positive result, CNNs, specifically CNNs such as the ones we have studied can detect the SZ effect in at least simple CMB, noise background. Reality is much complex, the galactic foregrounds constitute a open problem that future versions of this CNN will have to phase. The first one will be the division of the sky. For our validation/training sets, as we have homogeneity we randomly selected the sets. In realistic scenarios the galactic foreground are not homogeneous, we phase regions of the sky with different backgrounds (see figure 3.9), we then need to create the validation and train sets for each region, in multi-frequency this is a problem, as foregrounds behave differently. It is no obvious to define the regions.

Due to last time changes in the presentation of this work, we ran out of time to perform a second training, of either a different CNN on the same data or the same trained in a more realistic scenario. Nonetheless we decided to make a run of a more realistic data on our train, to see the behaviour and maybe any special feature. To do so we used one of our CMB + noise + SZ skies, (the ones used in the analysis of the method, not those used for the training. We detect a pretty good amount of founts given the

	0.05	0.1	0.2	0.3	0.4	0.5	0.6
N	620	529	441	391	351	312	261
N_{spur}	57.58%	56.14%	55.33%	54.22%	54.70%	55.13%	52.87%

Table 3.2: Number of prediction and spurious predictions using foreground data.

challenge. The fraction of spurious predictions seems to be nearly constant (compared to 250k catalogue).

Chapter 4

Conclusions and prospects

We have developed a multi-frequency based method for the detection of the Sunyaev-Zeldovich emission. This work should be seen as the first step of a more ambitious project. The we have started by training the network on a simplified simulated version of the sky, with no galactic foregrounds and assuming Gaussian instrumental noise, has been able to detect clusters of galaxies in the simulated skies. This result suggest that convolutional neural networks are suited for this kind of detection methods.

Even with a fewer detection number than the expected, the model have been able to fully use the frequency dependence of the SZ effect, characteristic that we expect exploit on following works of the SZ detection methods using machine learning. We have also learned about the absence of flux dependence, characteristic that might come from the labelling process, New implementations will be aware of such possibility.

To avoid the flux dependence problems we have suggested the use of variable-size labels, producing in this way a correlation of this flux, or brightness with the spatial correlation of the clusters, characteristic for what the CNN seems to be fully aware of. The SZ effect is one of the principal detection techniques for the large galaxy clusters. This work, fully developed, might help to create bigger and bigger catalogues of the SZ effects, this would contribute immensely on modern cosmology and astrophysics.

In addition to the SZ emission we know of backgrounds that also have characteristic frequency dependence. The success in the SZ detection will also provide new detection methods for apparently unrelated effects.

This job has covered the introduction of the method, the next steps would be, among many others, the comparison of different architectures of the CNNs on the same data, finding optimal structures which are not trivial. The most obvious next step would be the training of the same CNN, at different scenarios, getting into more and more realistic skies, but not directly, the implementation on small variation of already trained data can give us useful information. Ideally the last step would be the detection of SZ on CMB real data, the obvious candidate would be Planck data, ideally we could compare our results to their PSZ2, actually the greatest SZ emission sources catalogue.

A not trivial question will be the used of this CNNs on polarized data. The SZ emission is also polarized, maybe CNNs could be able to handle polarized and non polarized channels at the same size.

In conclusion these results suppose a first step in a very promising field that has been recently discovered and that have many surprised prepared for us, we will see.

Bibliography

[1] Planck Collaboration, N. Aghanim, Y. Akrami, *et al.*, "Planck 2018 results. I. Overview and the cosmological legacy of Planck," *Astronomy & Astrophysics*, vol. 641, A1, A1, Sep. 2020. DOI: 10.1051/0004-6361/201833880. arXiv: 1807.06205 [astro-ph.CO].

[2] A. A. Penzias and R. W. Wilson, "A Measurement of Excess Antenna Temperature at 4080 Mc/s.," *Astrophysical Journal*, vol. 142, pp. 419–421, Jul. 1965. DOI: 10.1086/148307.

[3] A. Challinor, "CMB anisotropy science: a review," in *Astrophysics from Antarctica*, M. G. Burton, X. Cui, and N. F. H. Tothill, Eds., vol. 288, Jan. 2013, pp. 42–52. DOI: 10.1017/S1743921312016663. arXiv: 1210.6008 [astro-ph.CO].

[4] D. Samtleben, S. Staggs, and B. Winstein, "The Cosmic Microwave Background for Pedestrians: A Review for Particle and Nuclear Physicists," *Annual Review of Nuclear and Particle Science*, vol. 57, no. 1, pp. 245–283, Nov. 2007. DOI: 10.1146/annurev.nucl.54.070103.181232. arXiv: 0803.0834 [astro-ph].

[5] M. Tegmark, "CMB mapping experiments: A designer's guide," *Physical Review D*, vol. 56, no. 8, pp. 4514–4529, Oct. 1997. DOI: 10.1103/PhysRevD.56.4514. arXiv: astro-ph/9705188 [astro-ph].

[6] G. de Zotti, M. Massardi, M. Negrello, and J. Wall, "Radio and millimeter continuum surveys and their astrophysical implications," *The Astronomy and Astrophysics Review*, vol. 18, no. 1-2, pp. 1–65, Feb. 2010. DOI: 10.1007/s00159-009-0026-0. arXiv: 0908.1896 [astro-ph.CO].

[7] J. Delabrouille, M. Betoule, J. .-B. Melin, *et al.*, "The pre-launch Planck Sky Model: a model of sky emission at submillimetre to centimetre wavelengths," *Astronomy & Astrophysics*, vol. 553, A96, A96, May 2013. DOI: 10.1051/0004-6361/201220019. arXiv: 1207.3675 [astro-ph.CO].

[8] C. Dickinson, *Cmb foregrounds - a brief review*, 2016. arXiv: 1606.03606 [astro-ph.CO].

[9] G. B. Rybicki and A. P. Lightman, *Radiative processes in astrophysics*. 1979.

[10] C. L. Bennett, R. S. Hill, G. Hinshaw, *et al.*, "First-year wilkinson microwave anisotropy probe (WMAP) observations: Foreground emission," *The Astrophysical Journal Supplement Series*, vol. 148, no. 1, pp. 97–117, Sep. 2003. DOI: 10.1086/377252. [Online]. Available: https://doi.org/10.1086/377252.

[11] A. Kogut, "Anomalous Microwave Emission," in *Microwave Foregrounds*, A. de Oliveira-Costa and M. Tegmark, Eds., ser. Astronomical Society of the Pacific Conference Series, vol. 181, Jan. 1999, p. 91. arXiv: astro-ph/9902307 [astro-ph].

[12] N. Boudet, H. Mutschke, C. Nayral, *et al.*, "Temperature Dependence of the Submillimeter Absorption Coefficient of Amorphous Silicate Grains," *Astrophysical Journal*, vol. 633, no. 1, pp. 272–281, Nov. 2005. DOI: 10.1086/432966.

[13] R. A. Sunyaev and Y. B. Zeldovich, "The Observations of Relic Radiation as a Test of the Nature of X-Ray Radiation from the Clusters of Galaxies," *Comments on Astrophysics and Space Physics*, vol. 4, p. 173, Nov. 1972.

[14] J. E. Carlstrom, G. P. Holder, and E. D. Reese, "Cosmology with the sunyaev-zel'dovich effect," *Annual Review of Astronomy and Astrophysics*, vol. 40, no. 1, pp. 643–680, 2002. DOI: 10.1146/annurev.astro.40.060401.093803. eprint: https://doi.org/10.1146/annurev.astro.40.060401.093803. [Online]. Available: https://doi.org/10.1146/annurev.astro.40.060401.093803.

[15] A. D. Dolgov, S. H. Hansen, S. Pastor, and D. V. Semikoz, "Spectral distortion of cosmic microwave background radiation by scattering on hot electrons: Exact calculations," *The Astrophysical Journal*, vol. 554, no. 1, pp. 74–84, Jun. 2001. DOI: 10.1086/321381. [Online]. Available: https://doi.org/10.1086/321381.

[16] M. Birkinshaw, "The Sunyaev-Zel'dovich effect," *Physics Reports*, vol. 310, no. 2-3, pp. 97–195, Mar. 1999. DOI: 10.1016/S0370-1573(98)00080-5. arXiv: astro-ph/9808050 [astro-ph].

[17] N. Itoh, Y. Kohyama, and S. Nozawa, "Relativistic corrections to the sunyaev-zeldovich effect for clusters of galaxies," *The Astrophysical Journal*, vol. 502, no. 1, pp. 7–15, Jul. 1998. DOI: 10.1086/305876. [Online]. Available: https://doi.org/10.1086/305876.

[18] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[19] M. Limon, E. Wollack, C. L. Bennett, *et al.*, *Wilkinson Microwave Anisotropy Probe (WMAP): Explanatory Supplement*, http://lambda.gsfc.nasa.gov/data/map/ doc/MAP_supplement.pdf, 2003.

[20] A. Zonca, L. Singer, D. Lenz, *et al.*, "Healpy: Equal area pixelization and spherical harmonics transforms for data on the sphere in python," *Journal of Open Source Software*, vol. 4, no. 35, p. 1298, Mar. 2019. DOI: 10.21105/joss.01298. [Online]. Available: https://doi.org/10.21105/joss.01298.

[21] K. M. Gorski, E. Hivon, A. J. Banday, *et al.*, "HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere," *The Astrophysical Journal*, vol. 622, no. 2, pp. 759–771, Apr. 2005. DOI: 10.1086/427976. [Online]. Available: https://doi.org/10.1086/427976.

[22] A. Lewis and S. Bridle, "Cosmological parameters from CMB and other data: A Monte Carlo approach," *Physical Review D*, vol. 66, p. 103 511, 2002. DOI: 10.1103/PhysRevD.66.103511. arXiv: astro-ph/0205436 [astro-ph].

[23] A. Lewis, A. Challinor, and A. Lasenby, "Efficient computation of CMB anisotropies in closed FRW models," *Astrophysical Journal*, vol. 538, pp. 473–476, 2000. DOI: 10.1086/309179. arXiv: astro-ph/9911177 [astro-ph].

[24] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications," *Astronomy and Computing*, vol. 27, pp. 130–146, 2019, ISSN: 2213-1337. DOI: https://doi.org/10.1016/j.ascom.2019.03.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213133718301392.

[25] N. Krachmalnicoff and M. Tomasi, "Convolutional neural networks on the HEALPix sphere: a pixel-based algorithm and its application to CMB data analysis," *Astronomy & Astrophysics*, vol. 628, A129, A129, Aug. 2019. DOI: 10.1051/0004-6361/201935211. arXiv: 1902.04083 [astro-ph.IM].

[26] B. Beckers and P. Beckers, "A general rule for disk and hemisphere partition into equal-area cells," *Computational Geometry*, vol. 45, no. 7, pp. 275–283, 2012, ISSN: 0925-7721. DOI: https://doi.org/10.1016/j.comgeo.2012.01.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925772112000296.

[27] Planck Collaboration, P. A. R. Ade, N. Aghanim, *et al.*, "Planck 2015 results. XXVII. The second Planck catalogue of Sunyaev-Zeldovich sources," *Astronomy & Astrophysics*, vol. 594, A27, A27, Sep. 2016. DOI: 10.1051/0004-6361/201525823. arXiv: 1502.01598 [astro-ph.CO].

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[29] M. Abadi, A. Agarwal, P. Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[30] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, 2015, vol. 25.

[31] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79.

[32] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[33] Y. Lu, *Food image recognition by using convolutional neural networks (cnns)*, 2019. arXiv: 1612.00983 [cs.CV].

[34] B. D., *A deep learning approach for the detection of point sources in the cosmic microwave background*, 2019.

[35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241, ISBN: 978-3-319-24574-4.